

UNIVERSIDADE DE LISBOA



**Introdução à Programação e à Robótica Educativa no Ensino Básico
através da metodologia de Aprendizagem Baseada em Problemas**

João Paulo Nogueira de Castro Freitas

Mestrado em Ensino da Informática

Relatório da Prática de Ensino Supervisionada orientado pela Professora Doutora
Neuza Pedro e pelo Professor Doutor Luís Moniz

2022

Agradecimentos

Em primeiro lugar quero agradecer à Professora Doutora Neuza Pedro, orientadora da Prática de Ensino Supervisionada do Instituto de Educação, pelos contributos, correções e sugestões, especialmente durante o desenvolvimento deste documento no qual foi sempre pautada por intervenções pertinentes, assertivas e de grande rigor científico.

Em segundo lugar, ao Professor Doutor Luís Moniz, orientador da Faculdade de Ciências, por toda a orientação, comentários e sugestões ao longo do processo, e disponibilidade sempre que foi necessário.

Em terceiro lugar, ao Professor Cooperante Paulo Torcato, por toda a sapiência demonstrada e partilhada na área da Robótica Educativa, fornecendo sempre sugestões pertinentes sobre a melhor forma de aplicar os exercícios, procurando sempre exemplificar as melhores formas de resolver os problemas, sugerindo alternativas quando necessário, apoiando na orientação da avaliação dos alunos e sempre disponível para uma troca de ideia mesmo estando fora do país em trabalho, esteve sempre presente e preocupado com o desenrolar da intervenção.

Em quarto lugar, aos Professores Doutores João Piedade e Nuno Dorotea do Instituto de Educação, por todo o apoio, aconselhamento e acompanhamento que incansavelmente me deram ao longo do Mestrado.

Aos colegas que durante estes dois anos foram amigos, companheiros, críticos e partilharam inúmeros momentos de experiências e saberes, em especial ao Amâncio Ferreira, que desde dos primeiros dias foi colega de grupo em inúmeros trabalhos e sempre esteve lá para partilhar uma ideias ou trocar umas opiniões.

À minha família, à minha esposa Aurora e ao meu filho Dinis pela paciência, compreensão e por tolerarem as minhas ausências inúmeras vezes em detrimento do Mestrado.

A todos um sincero, muito obrigado.

Resumo

Este relatório descreve o exercício da Prática de Ensino Supervisionada realizada na Escola Secundária da Portela, do concelho de Loures, no âmbito do Mestrado em Ensino de Informática, do Instituto de Educação da Universidade de Lisboa. Esta intervenção decorreu numa turma do 9º ano de escolaridade, durante o 2.º e 3.º período do ano letivo 2021/2022. Foram lecionadas 10 aulas no total de 500 minutos letivos na unidade temática de “Introdução à Robótica”. Nesta intervenção foi implementado o cenário aprendizagem “*Smart City*”, dedicado à programação de robots num contexto urbano para assim sensibilizar os alunos para as questões da gestão da mobilidade nas cidades. A metodologia de aprendizagem selecionada foi a Aprendizagem Baseada em Problemas (ABP), recorrendo-se à robótica educativa como estratégia de ensino para a aprendizagem inicial da programação no ensino básico e também como estratégia promotora de competências cognitivas e sociais, por forma a estruturar os conhecimentos dos alunos relativos à temática abordada. Foi ainda trabalhada uma dimensão de investigação onde se procurou responder à seguinte questão de investigação: Q1: *A aprendizagem da programação de robots será idêntica nas raparigas e nos rapazes?* Os resultados mostraram que a questão foi respondida em sentido favorável na medida em que não se registaram diferenças significativas entre o grupo de raparigas e o grupo de rapazes, tanto para a totalidade do teste de conhecimentos como para as diferentes dimensões em estudo: algoritmia, robótica e programação. Os resultados relativos à intervenção, centrados nas aprendizagens realizadas pelos alunos (pré-teste e pós-teste) e na sua apreciação relativamente ao desempenho do professor, demonstraram igualmente que esta intervenção foi bem-sucedida face aos objetivos definidos para o cenário implementado.

Palavras-chave: Aprendizagem Baseada em Problemas; Introdução à Programação; Pensamento Computacional; Programação por Blocos; Robótica Educativa.

Abstract

This report describes the Supervised Teaching Practice carried out at the Secondary School of Portela, in the municipality of Loures, as part of the Master's Degree in Computer Science Teaching, of the Institute of Education of the University of Lisbon. This intervention took place in a 9th grade class, during the 2nd and 3rd terms of the school year 2021/2022. During 500 minutes in 10 lessons were taught about the thematic unit of "Introduction to Robotics". In this intervention the learning scenario "Smart City" was implemented, dedicated to the programming of robots in an urban context to raise students' awareness of mobility issues and mobility management in cities. The learning methodology selected was Problem-Based Learning (PBL), using educational robotics as a teaching strategy for the initial learning of programming in primary education, and also as a strategy to promote cognitive and social skills, in order to structure the students' knowledge regarding the theme addressed. A research dimension was also developed to answer the following research question: Q1: Is the learning of robot programming identical for girls and boys? The results have showed that the question was favourably answered insofar as there were no significant differences between the group of girls and the group of boys, both for the totality of the knowledge test and for the different dimensions under study, which were: algorithm, robotics and programming. The results regarding the intervention, focused on the learning achieved by the students (pre-test and post-test) and on their appreciation of the teacher's performance, that have showed that this intervention was successful regarding the objectives defined for the implemented scenario.

Keywords: Problem-Based Learning; Introduction to Programming; Computational Thinking; Block Programming; Educational Robotics.

Índice

1.	Introdução	1
2.	Caraterização do Contexto	3
2.1.	Contexto Social	3
2.2.	Contexto Escolar	4
2.2.1.	Agrupamento de Escolas de Portela e Moscavide	5
2.2.2.	Escola Secundária da Portela (Arco-Íris).....	6
2.2.3.	Turma	7
2.2.4.	Horário da Turma.....	8
2.2.5.	Sala de Aula	9
2.3.	Métodos de Análise Diagnóstica.....	10
3.	Enquadramento Curricular	11
3.1.	Unidade Didática: Introdução à Robótica	11
3.2.	Unidade Temática: Introdução à Programação com o recurso à Robótica .	12
3.3.	Conceitos científicos	14
3.3.1.	Ações.....	14
3.3.2.	Algoritmos	15
3.3.3.	Estruturas de Controle.....	15
3.3.3.1	Estruturas de Decisão ou Seleção	15
3.3.3.2	Estruturas de Repetição ou Ciclos	18
3.3.4.	Eventos.....	19
3.3.5.	Programação de Sensores.....	19
3.3.6.	Operadores Relacionais, Lógicos e Aritméticos	20
3.3.7.	Tipos de Dados.....	20
3.3.8.	Variáveis	21
4.	Problematização da Temática a Lecionar	23

4.1.	Pensamento Computacional	24
4.2.	As Principais Dificuldades no Ensino da Programação	25
4.2.1.	A Complexidade da Programação.....	25
4.3.	Disciplina de Introdução à Robótica	26
4.3.1.	A Robótica Educativa	27
5.	Intervenção Pedagógica	29
5.1.	Observação de aulas	29
5.2.	Cenário de Aprendizagem	31
5.3.	Objetivos de Aprendizagem	33
5.4.	Metodologia de avaliação das aprendizagens.....	33
5.5.	Metodologia de Aprendizagem Baseadas em Problemas.....	35
5.6.	Atividades.....	35
5.7.	Recursos	36
5.8.	Operacionalização das atividades letivas	37
5.8.1.	1.ª e 2.ª aula - 23.02.2022.....	39
5.8.2.	3.ª e 4.ª aula - 02.03.2022.....	40
5.8.3.	5.ª e 6.ª aula - 16.03.2022.....	41
5.8.4.	7.ª e 8.ª aula - 30.03.2022.....	42
5.8.5.	9.ª e 10.ª aula - 04.05.2022.....	43
6.	Dimensão Investigativa.....	45
6.1.	Instrumentos e procedimentos de Recolha de Dados	46
6.2.	Resultados	47
6.2.1.	Análise comparativa pré-teste e pós-teste	47
6.2.2.	Análise comparativa por género.....	50
6.3.	Componente Investigativa	51
6.4.	Avaliação das Aula Lecionadas	52
7.	Conclusão.....	57

7.1. Balanço Reflexivo	58
8. Referências	61
9. Anexos	65

Índice de Figuras

Figura 2.1 - Escola Secundária da Portela (fonte: Google Maps, 2022).....	6
Figura 2.2 - Horário da turma 9.º B	8
Figura 2.3 - Sala de aula - C3.....	9
Figura 3.1 - Taxonomia de Bloom - revista (fonte: Anderson & Khathwohl, 2001). 14	
Figura 3.2 – Diagrama sintático da estrutura if... then... else... (fonte: Porto Editora, 2022)	16
Figura 3.3 – Diagrama representativo da instrução ou estrutura if... then... else... (fonte: Porto Editora, 2022)	17
Figura 3.4 - Representação esquemática - fuxograma de uma estrutura de decisão if... then... else... (fonte: Porto Editora, 2022).....	18
Figura 4.1 - Fases do Pedagogical Content Knowledge (fonte: Shulman, 1987)	23
Figura 5.1 - Princípios orientadores para o desenho de cenários de aprendizagem (fonte: Matos, 2014).....	32
Figura 5.2 - Esquema da Aprendizagem Baseada em Problemas (fonte: Piedade et al., 2017)	35
Figura 5.3 - Posto de trabalho dos grupos.....	37
Figura 5.4 - Grupos de trabalho a programar Robots.....	39
Figura 5.5 - Grupo de raparigas a programar o Robot	40
Figura 5.6 - Aluno a testar Robot no cenário	41
Figura 5.7 - Utilização de diferentes Robots no cenário.....	42
Figura 5.8 - Simulação de obstáculos no cenário.....	43
Figura 6.1 - Gráfico 1. Dimensão da Prática Letiva dos Alunos	53
Figura 6.2 - Gráfico 2. Dimensão do Processo de Ensino e Aprendizagem	54
Figura 6.3 - Gráfico 3. Dimensão de Autoavaliação dos Alunos.....	55

Índice de Tabelas

Tabela 2.1- Escolas do Agrupamento de Escolas de Portela e Moscavide.....	4
Tabela 3.1- Operadores relacionais, lógicos e aritméticos (fonte: Santos, 2006)	20
Tabela 6.1 - Resultado da Estatística Descritiva do Teste de Wilcoxon.....	48
Tabela 6.2 - Resultado do resumo de Teste de Hipótese (pontos).....	48
Tabela 6.3 – Resultado do resumo de Teste de Hipótese (algoritmia).....	48
Tabela 6.4 - Resultado do resumo de Teste de Hipótese (robótica).....	49
Tabela 6.5 - Resultado do resumo de Teste de Hipótese (programação).....	49
Tabela 6.6 - Resultado da Estatística Descritiva das médias (Teste-T amostras independentes).....	50
Tabela 6.7 - Resultado do Teste Não Paramétrico (Mann-Whitney).....	51

Índice de Anexos

Anexo A. Mapa de Conceitos Científicos da Programação de Robots mBot.....	65
Anexo B. Grelha de observação – Estratégias de Ensino - Professor.....	66
Anexo C. Grelha de observação – Dinamização da Aula - Alunos.....	67
Anexo D. Cenário de Aprendizagem – Smart City.....	68
Anexo E. Plano de aula n.º 1 e 2.....	79
Anexo F. Plano de aula n.º 3 e 4.....	80
Anexo G. Plano de aula n.º 5 e 6.....	81
Anexo H. Plano de aula n.º 7 e 8.....	82
Anexo I. Plano de aula n.º 9 e 10.....	83
Anexo J. Plataforma Google Classroom da disciplina.....	84
Anexo K. Apresentação eletrónica – aulas n.º 1/2.....	86
Anexo L. Apresentação eletrónica – aulas n.º 3/4.....	101
Anexo M. Apresentação eletrónica – aulas n.º 5/6.....	112
Anexo N. Apresentação eletrónica – aulas n.º 7/8.....	123
Anexo O. Apresentação eletrónica – aulas n.º 9/10.....	133
Anexo P. Autorização para realização investigação (Direção do AEPM).....	141
Anexo Q. Autorização para realização investigação (Encarregados de Educação). 142	
Anexo R. Teste de Normalidade (Kolmogorov-Smirnov/Shapiro-Wilk).....	143
Anexo S. Teste de Homogeneidade de Variância (Estatística de Levene).....	144
Anexo T. Ficha de Avaliação Inicial (pré-teste).....	145
Anexo U. Ficha de Avaliação Final (pós-teste).....	153
Anexo V. Apresentação eletrónica - Algoritmos.....	162
Anexo W. Rubrica de avaliação formativa das atividades.....	180
Anexo X. Rubrica de avaliação sumativa da atividade final.....	181
Anexo Y. Inquérito aos alunos sobre a Prática de Ensino Supervisionada.....	182
Anexo Z. Guião de apoio 01 – Blink.....	190
Anexo AA. Guião de apoio 02 – Alternat Flashing.....	192
Anexo BB. Guião de apoio 03 – Mixed Color.....	194
Anexo CC. Guião de apoio 04 – Random Color.....	197
Anexo DD. Guião de apoio 05 – Button Play.....	199
Anexo EE. Guião de apoio 06 – Play Music.....	201
Anexo FF. Guião de apoio 07 – Ambulance Sound.....	203

Anexo GG. Guião de apoio 08 –Sound Gradient.....	205
Anexo HH. Guião de apoio 09 – Brightness Gradient.....	207
Anexo II. Guião de apoio 10 – Fire Engine Sound	209
Anexo JJ. Guião de apoio 11 – Light Sensor Value	211
Anexo KK. Guião de apoio 12 – Infrared Communication	213
Anexo LL. Guião de apoio 13 – Keyboard Control.....	215
Anexo MM. Guião de apoio 14 – M Trail	217
Anexo NN. Guião de apoio 15 – S Trail.....	219
Anexo OO. Guião de apoio 16 – Ambulance	221
Anexo PP. Guião de apoio 17 – Ultra Sonic Sensor Value	222
Anexo QQ. Guião de apoio 18 – Auto Stop.....	223
Anexo RR. Guião de apoio 19 – Obstacle Avoidance	224
Anexo SS. Guião de apoio 20 – Line Follower Value.....	226
Anexo TT. Guião de apoio 21 – Let Me Down	228
Anexo UU. Guião de apoio 21 – Cliff Detection.....	230
Anexo VV. Guião de apoio 23 – Simple Follow Program	231

Lista de abreviaturas

AEPM - Agrupamento de Escolas Portela e Moscavide

APB - Aprendizagem Baseada em Problemas

ASE - Ação Social Escolar

CCEMS - Centro de Competências Entre Mar e Serra

DEP - Dificuldades no Ensino da Programação

ERASMUS - European Region Action Scheme for the Mobility of University Students

FCUL - Faculdade de Ciências da Universidade de Lisboa

FIP - Formação Inicial de Professores

IE - Instituto de Educação

LED - Light-Emitting Diode

LVPB - Linguagem Visual de Programação por Blocos

NE - Necessidades Educativas

OCDE - Organização para a Cooperação e Desenvolvimento Económico

PC - Pensamento Computacional

PE - Professor Estagiário

PES - Prática de Ensino Supervisionada

PGC - Plataforma Google Classroom

PIR - Passive Infrared Sensor

RE - Robótica Educativa

RED - Recursos Educativos Digitais

RGB - Red, Green, Blue

SIMA - Sistema Interativo de Monitorização das Aprendizagens

STEM - Science, Technology, Engineering e Mathematics

1. Introdução

Este documento foi proposto como trabalho final da disciplina de Introdução à Prática Profissional IV, e consiste na apresentação de um relatório descritivo do projeto de intervenção que integrou a prática pedagógica no âmbito da Prática de Ensino Supervisionada (PES) do Mestrado em Ensino de Informática.

Esta intervenção foi efetuada na unidade didática de Introdução à Robótica, na turma C do 9.º ano de escolaridade na Escola Secundária da Portela (Arco-íris), do Agrupamento de Escolas de Portela e Moscavide, e teve como unidade temática de intervenção a “Introdução à Programação dos Robots” (*mBot*), através da linguagem de programação por blocos do *mBlock*.

A estrutura deste documento é constituída por sete capítulos, finda esta introdução (capítulo um), temos o capítulo dois no qual é feita a caracterização do contexto tanto social como escolar e apresento a análise diagnóstica efetuada.

No terceiro capítulo abordo o enquadramento curricular e didático da turma descrevendo ambas as unidades bem como abordando os conceitos científicos que serão utilizados na intervenção.

No quarto capítulo apresento a problematização da temática a lecionar, na qual aqui reflito sobre as temáticas conexas à intervenção, como, o pensamento computacional, as dificuldades e a complexidade do ensino da programação, e por último a análise da disciplina da intervenção: Introdução à Robótica.

No quinto capítulo descrevo a intervenção pedagógica com todas as suas características: a observação das aulas efetuadas, o cenário e os objetivos de aprendizagem, a metodologia de avaliação e a metodologia de aprendizagem baseada em problemas, as atividades e recursos, e por último a operacionalização das atividades letivas.

No sexto capítulo descrevo sucintamente o que foi a dimensão investigativa a realizar no decurso da intervenção.

No sétimo e último capítulo faço a conclusão de todo o trabalho realizado.

2. Caracterização do Contexto

A implementação do currículo, tem questões inerentes ao contexto que ganham importância na medida em que condicionam a própria gestão curricular, esta deverá ter em atenção fatores contextuais como o meio sociocultural dos alunos, as condições das infraestruturas da escola, as expectativas dos atores educativos, entre outros fatores.

Assim, a necessidade de se conhecer o contexto para poder organizar o desenvolvimento curricular, é apresentada neste capítulo, desde o nível social, escolar, de turma e de sala de aula, onde se desenvolveu a prática de ensino supervisionada.

Solicitei, através do Professor Cooperante alguns documentos de caracterização da turma para perceber melhor as características dos alunos que iria acompanhar e realizar a caracterização da turma neste capítulo.

2.1. Contexto Social

O Agrupamento de Escolas de Portela e Moscavide (AEPM) está integrado na Área Metropolitana de Lisboa, e pertence ao concelho de Loures, mais precisamente na União de Freguesias de Moscavide e Portela (agregação de freguesias efetuada em 2013). A freguesia de Moscavide foi constituída em 23 de março de 1928, pelo Decreto n.º 1522, e a freguesia da Portela em 4 de outubro de 1985, pela Lei n.º 111/85.

Na freguesia da Portela situa-se uma mancha urbana com cerca de 30.000 habitantes pertencentes a um estrato sociocultural médio e médio-alto.

Fazem igualmente parte da freguesia da Portela alguns bairros periféricos, onde se entrecruzam populações de etnias e religiões diversas.

Na freguesia de Moscavide reside um estrato populacional sociocultural médio-baixo e é nesta parte da freguesia que residem o maior número de imigrantes dos países do leste da Europa e Brasil.

Ultimamente nota-se um certo rejuvenescimento nas duas freguesias, resultante do regresso de alguma descendência à Portela e da existência de um grande número de imigrantes dos países do leste da Europa em Moscavide, com isto novas preocupações surgem para as escolas do Agrupamento.

2.2. Contexto Escolar

O Agrupamento de Escolas de Portela e Moscavide iniciou a sua atividade no ano letivo 2003/2004, e era composto por 4 escolas: Escola-sede; Escola EB 2,3 Gaspar Correia; Escola Básica do 1º Ciclo e J.I. da Portela; Escola Básica do 1º Ciclo de Moscavide e a Escola Básica EB 1 Dr. Catela Gomes.

A partir do ano letivo de 2010/2011 este Agrupamento passou a integrar a Escola Secundária da Portela, que passou a Escola-sede, e continuou a designar-se por Agrupamento de Escolas de Portela e Moscavide.

Atualmente o agrupamento é composto por cinco estabelecimentos de ensino, sendo eles:

Tabela 2.1- Escolas do Agrupamento de Escolas de Portela e Moscavide

Ano letivo	Escolas do agrupamento	Nível de ensino
2003/2004	E.B. 1/JI da Portela	Educação Pré-Escolar e 1.º ciclo
	E.B. 1/JI Quinta da Alegria	
	E.B. 1/JI da Dr. Catela Gomes	1.º Ciclo
	E.B. 2,3 Gaspar Correia	2.º e 3.º Ciclos
2010/2011	Escola Secundária da Portela (sede)	3.º Ciclo e Secundário

A população das freguesias da Portela e de Moscavide é servida por três escolas do 1º Ciclo – EB 1/JI da Portela, EB 1/JI Quinta da Alegria e EB 1 Dr. Catela Gomes.

Os alunos das três escolas do 1º Ciclo pertencentes ao Agrupamento são, normalmente, encaminhados para a Escola EB 2,3 Gaspar Correia.

No final do 2º ciclo os alunos permanecem na Escola EB 2,3 Gaspar Correia ou são encaminhados para a Escola Secundária da Portela, uma vez que não há capacidade para o 3º ciclo estar todo na Escola Secundária da Portela.

A população escolar da Escola Secundária Arco-Íris é constituída atualmente pelo seguinte número de alunos: 449 do 3.º ciclo do ensino básico, integrando 17 turmas; 85 do ensino profissional, integrando 3 turmas e 555 dos cursos científico-humanísticos do ensino secundário, integrando 20 turmas.

A orientação e desenvolvimento do processo de ensino e de aprendizagem é assegurada por 100 docentes, por sua vez, o pessoal não docente conta com 33 colaboradores.

2.2.1. Agrupamento de Escolas de Portela e Moscavide

Após a análise do Portal do Agrupamento e da leitura de vários documentos estruturantes, como o Projeto Educativo 2020/2023 (AEPM a, 2020), o Regulamento Interno 2020/2024 (AEPM b, 2020) e o Documento Orientador para a organização do ano letivo 2021/2022 (AEPM, 2021), efetuou a seguinte análise descritiva.

O Agrupamento tem como missão dar resposta às preocupações dos alunos sobre as profissões que podem vir a exercer futuramente, através da reflexão, ou seja, os intervenientes no processo de ensino, têm que pensar na Escola que querem para os seus alunos. Num quadro de autonomia, de flexibilidade curricular, de interdisciplinaridade e de projetos, serão necessárias ideias inovadoras e debate, para que se possa levar em frente os pressupostos de uma escola integradora e virada para o futuro (N. Reis, 2019).

O Agrupamento possui vários serviços especializados de apoio educativo, tais como: a Educação Especial - Apoios Especializados; a Estrutura de Orientação Educativa/Técnico-Pedagógica; o Serviço de Psicologia e Orientação; o Apoio Educativo (Socioeducativo) – 1º Ciclo e os Apoios Pedagógicos Acrescidos (APA's) 2.º/3.º Ciclos e Ensino Secundário. Estes serviços destinam-se a promover a criação de condições que assegurem a plena integração escolar e social dos alunos e, para isso, deve ser conjugada a sua atividade com as estruturas de orientação educativa.

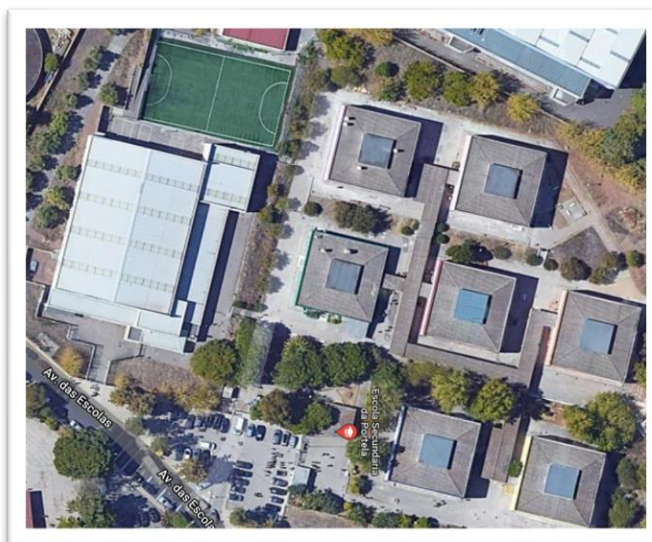
Também conta com algumas parcerias como o Município (no âmbito da Ação Social Escolar e manutenção de edifícios e recreios escolares), Bombeiros Voluntários de Moscavide e Portela, Empresas locais como a GesLoures (competências complementares e preparação para a vida ativa de alunos) e protocolos com várias instituições designadamente o Centro de Recursos para a Inclusão da Cercipóvoa (no âmbito da Educação Especial).

2.2.2. Escola Secundária da Portela (Arco-Íris)

A Escola foi contruída em 1988, e é constituída por 8 blocos pintados, cada um com uma cor do arco-íris, sendo eles:

- Bloco 1 (verde): Refeitório, Bar dos Alunos e Papelaria;
- Bloco 2 (sem cor): Pavilhão Gimnodesportivo;
- Bloco A (azul): Direção, Serviços Administrativos, Biblioteca Escolar, Sala de Professores, Sala de Diretores de Turma, Gabinete Médico e Serviços de Reprografia;
- Bloco B (amarelo): nove salas, disciplinas ligadas às Artes Visuais;
- Bloco C (laranja): treze salas, 3.º Ciclo;
- Bloco D (vermelho): dez salas de aula, 3.º Ciclo, Sala de Teatro e Auditório da Escola;
- Bloco E (violeta): onze salas de aula, Ensino Secundário e Laboratórios de Biologia;
- Bloco F (anil): nove salas de aula, Ensino Secundário e os Laboratórios de Química.

Figura 2.1 - Escola Secundária da Portela (fonte: Google Maps, 2022)



Ao nível do Ensino Secundário, a escola tem como oferta educativa os Cursos Científico-Humanísticos: Ciências e Tecnologias; Ciências Socioeconómicas, Artes Visuais e Línguas e Humanidades. Ao nível do Ensino Profissional tem o Curso Profissional de Técnico de Desporto.

Os alunos podem usufruir de várias atividades inseridas em clubes: Ciência; Robótica; Multimédia; Teatro e Matemática; no Desporto Escolar e em variados eventos que se realizam ao longo do ano, especialmente no dia do agrupamento. Estes clubes operam em espaços próprios, criados e equipados em função das atividades específicas de cada um deles.

2.2.3. Turma

A turma do 9.º C era composta por 25 alunos, sendo 17 do género masculino e 8 do género feminino com uma média de idades de 14 anos (68% dos alunos tem 14 anos). Os alunos são todos de nacionalidade portuguesa, a maioria 63% habita na união de freguesias de Portela e Moscavide, sendo que os restantes 27% habitam em freguesias adjacentes, contudo todos os alunos habitam com os respetivos progenitores.

Os encarregados de educação são maioritariamente as mães (80% dos alunos), 59% dos encarregados de educação possuem habilitação ao nível do ensino superior, 29% indicam ter habilitações ao nível do ensino secundário, 4% indicam ter habilitações ao nível do 1.º ciclo e 8% não indicam qualquer habilitação dos encarregados de educação.

Relativamente à Ação Social Escolar (ASE), 88% indicam não usufruir, somente 12% dos alunos usufruem, sendo que 8% usufrui de escalão A e 4% de escalão B. Não constavam alunos com Necessidades Educativas (NE).

Estes alunos pertencem a um extrato social médio ou médio/alto, os pais valorizam a escola como instituição formativa e fazem um acompanhamento sistemático e constante da vida escolar dos seus educandos, as condições que estas famílias usufruem permitem que os alunos sejam participativos e organizados com desempenhos bastante razoáveis.

Consultada a caracterização fornecida pela diretora de turma no início do ano letivo constatei que 83% dos alunos não apresentam retenções no seu percurso escolar sendo que os restantes 17% apresentam uma ou mais retenções.

Após a reunião do conselho de turma do primeiro período constatei que 59% dos alunos não apresentam qualquer nível negativo, dos restantes 41% que obtiveram níveis negativos só 20% é que apresentam 3 ou mais níveis negativos, sendo que a

média das classificações da turma situa-se nos 3,4 valores o que foi considerado pelo Conselho de Turma como uma boa média.

O comportamento global da turma foi caracterizado como sendo uma turma agitada, barulhenta, contudo participativa, por último de salientar que quatro alunos da turma integraram um projeto Erasmus+ (*New Generation Schools in the Light of Education 4.0*) no qual o Agrupamento participou.

2.2.4. Horário da Turma

O horário da disciplina de Introdução à Robótica foi de um tempo letivo semanal de 50 minutos às quartas-feiras das 9h55 às 10h45.

Para a intervenção foram necessárias 10 aulas de 50 minutos, no entanto, o Professor Cooperante estabeleceu uma permuta com a professora de Geografia e foram garantidas as aulas de 100 minutos, às quartas-feiras das 8h55 às 9h45 e das 9h55 às 10h45, sendo assim só necessárias 5 aulas para realizar a intervenção, esta alteração foi comunicada à direção deste Agrupamento de Escolas e aos respetivos encarregados de educação para autorização de frequência.

Figura 2.2 - Horário da turma 9.º B

Ano letivo: 2021-22

Tempos	Segunda	Sala	Terça	Sala	Quarta	Sala	Quinta	Sala	Sexta	Sala
08:00 - 08:50			História	D08	C. Naturais	D08	História	D08	C. Naturais	D08
08:55 - 09:45	MAT	D08	Ed. Física	GIN	Geografia	D08	Inglês	D08	Francês	D08
09:55 - 10:45			Francês	D08	Robótica	C03	C. F.Química	D08	Português	D08
10:55 - 11:45	Inglês	D08	Português	D08	Ed. Física	GIN	MAT	D08		
12:00 - 12:50	CFQ T1 CN t2	F03 C06							TIC 1ºsem T2	C04
12:55 - 13:45	CFQ T2 CN T1	F03 C06					TIC 1ºsem T1	C04	Cid 2º sem	D08
13:50 - 14:40			Geografia	D08						
14:45 - 15:35	Ed. Física	GIN	C. F.Química	D08						
15:45 - 16:35	Ed. Visual	D08								
16:40 - 17:30										
17:35 - 18:25										

Entrada em vigor: 15/09/2021

Data de Validade: 31 de Agosto de 2022

2.2.5. Sala de Aula

A sala onde ocorreram as atividades letivas da disciplina de Introdução à Robótica foi a sala 3 do bloco C (intitulada C3), no bloco cor de laranja, tendo esta o nome do projeto LAB aTTitude 3D e está localizada no rés-do-chão do mesmo pavilhão. Esta possui uma boa acessibilidade e mobilidade, sendo uma sala ampla com bastante luz natural e uma boa sonoridade.

Figura 2.3 - Sala de aula - C3



A disposição da sala é configurável, pois esta não tem mobiliário convencional, mas sim mobiliário modular que pode ser deslocado rapidamente e alterando a configuração da organização da sala em poucos segundos. O mobiliário da sala é composto pelos seguintes itens:

- 4 Cadeiras com rodas e palmatória;
- 5 Pufs de esponja;
- 2 anfiteatros com rodas (nos quais tem armazenamento para mochilas debaixo da primeira escada, e ainda fichas para carregar equipamentos);
- 12 Mesas cor de laranja em plástico com rodas;
- 18 Cadeiras de plástico ergonómicas;
- 3 Mesas com tampo de madeira;
- 6 Lousas brancas com marcadores;
- 1 Secretária para o professor.

Existe ainda à disposição um quadro interativo *Promethean ActiveBoard* com suporte com rodas, para ser mais fácil a movimentação pela sala.

Esta sala possui ainda uma arrecadação para armazenamento dos equipamentos tecnológicos (tablets, robots, computadores, ferramentas, kits tecnológicos), bem como outros materiais de apoio à disciplina.

Todo este mobiliário está em excelentes condições de utilização e é amovível o que facilita bastante a modificação da disposição da sala, permitindo dispor facilmente de áreas diferentes de trabalho, tanto individuais e em pequenos ou grandes grupos.

Quanto aos equipamentos tecnológicos esta sala dispõe de:

- 1 Computador fixo, ligado ao quadro interativo;
- 10 Computadores portáteis;
- 15 Tablets 10” Android;
- 10 Kit do Robot Educativo *mBot* v1;
- 20 Kits do Micro:Bit v1;
- 5 Kits do Micro:Bit v2;
- 1 Kit Arduino;
- 2 Impressoras 3D.

Estes equipamentos, encontram-se em boas condições físicas e de utilização.

Este espaço foi pensado para proporcionar condições de trabalho tanto individuais como colaborativas, possibilitando uma cultura de reciprocidade na aprendizagem e assim possibilitando práticas de partilha e colaboração entre pares.

2.3. Métodos de Análise Diagnóstica

A análise diagnóstica da turma foi efetuada a partir dos documentos fornecidos pela Diretora de Turma, estes resultaram no essencial da aplicação da ficha de caracterização da turma no início do ano letivo, também foi disponibilizada uma tabela com o resumo do aproveitamento da turma, na qual constavam: retenções; necessidades educativas; ação social escolar: quadros de valor e de excelência; apoios pedagógicos; a pedagogia diferenciada; as acomodações curriculares dos alunos e ainda a escolha da disciplina opcional.

3. Enquadramento Curricular

A intervenção supervisionada foi realizada no âmbito da disciplina de Introdução à Robótica, onde abordei os conceitos básicos da programação de robots, tais como, as ações, as estruturas de controle e de seleção, os eventos, as instruções de repetição (ciclos), a programação de sensores e atuadores, os operadores (relacionais, lógicos e aritméticos), os tipos de dados e as variáveis.

3.1. Unidade Didática: Introdução à Robótica

Esta disciplina faz parte da oferta de escola prevista na oferta formativa ao nível do ensino básico (7.º, 8.º e 9.º ano de escolaridade) neste Agrupamento de Escolas, esta oferta foi idealizada com uma visão para proporcionar um destaque da programação de computadores e da robótica, procurando assim oferecer práticas de ensino mais inovadoras e atuais e com grande procura no mercado de trabalho.

Esta disciplina foi pensada para concretizar os seguintes objetivos:

- Introduzir aos alunos os conceitos fundamentais de programação;
- Introduzir aos alunos conceitos de robótica;
- Estimular a criatividade e o pensamento crítico;
- Promover a interdisciplinaridade;
- Utilizar conceitos de outras áreas do conhecimento para a aplicação na resolução de problemas;
- Estimular os alunos na execução de atividades experimentais;
- Promover a investigação e resolução de problemas na procura da melhor solução de programação dos robots;
- Procurar que o aluno se torne agente ativo e autónomo na procura do seu próprio conhecimento;
- Promover um espaço para gerir comportamentos e emoções, obter responsabilidade e lidar com a frustração.

Para o corrente ano letivo foram planificadas para esta disciplina 34 aulas de cinquenta minutos, sendo a sua estrutura modular a seguinte:

- 1. Algoritmia e Resolução de Problemas;**
- 2. Introdução à Robótica;**

- a. O Kit Educativo *mBot Makeblock* e a sua montagem;

3. Introdução à Programação com o recurso à Robótica;

- a. Componentes de um programa;
- b. Carregamento do programa *mBlock* para o robot *mBot*;
- c. Utilização de atuadores de movimento;
- d. Estruturas de controlo, repetição e seleção;
- e. Utilização de atuadores de som e luz;
- f. Utilização do sensor de proximidade;
- g. Utilização de variáveis;
- h. Utilização de sensor de seguimento de linha;
- i. Tipos de dados;
- j. Tipos de operadores e expressões lógicas.

De referir que o Professor Cooperante lecionou os módulos 1 e 2, sendo que a minha intervenção incidiu nos conteúdos curriculares no módulo 3 alínea c) até à alínea j).

3.2. Unidade Temática: Introdução à Programação com o recurso à Robótica

A minha intervenção ocorreu dentro da unidade temática de “Introdução à Programação com o recurso à Robótica”, com os conteúdos curriculares referidos anteriormente, assim desta forma, pretendi que os alunos no final fossem capazes de:

- Identificar os componentes estruturais da programação;
- Identificar diferentes operadores e funções pré-definidas;
- Usar os diferentes operadores aritméticos, lógicos e relacionais;
- Compreender o funcionamento das estruturas de seleção e repetição;
- Identificar diferentes tipos de dados;
- Compreender o funcionamento das variáveis;
- Criar sequências de instruções que envolvam a seleção e a repetição;
- Criar programas que envolvam a utilização de variáveis e funções;
- Identificar e corrigir erros existentes na programação;
- Aplicar os conceitos base da programação na resolução de problemas.

A interdisciplinaridade é o processo de ligação entre as diferentes disciplinas, assim ao efetuar as planificações das aulas de intervenção, verifiquei que existiam pontos de convergência com outras disciplinas, podendo assim abordar diversos conteúdos em conjunto, proporcionando aos alunos uma melhor compreensão e aprendizagem.

Os domínios escolhidos foram os seguintes:

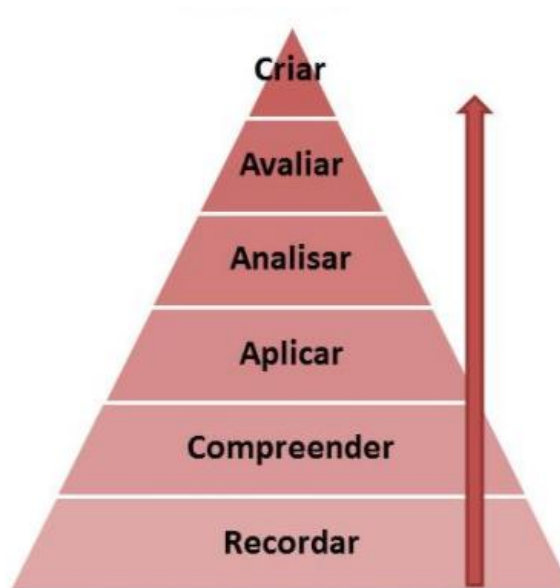
- Matemática (domínios: Números e Operações; Resolução de problemas, Raciocínio e Comunicação);
- Físico-Química (domínios: Forças, movimentos e energia);
- TIC (domínios: Colaboração em ambientes fechados; Exploração de ambientes computacionais);
- Inglês (domínios: Competência Comunicativa);
- Cidadania e Desenvolvimento (domínios: Desenvolvimento Sustentável; Educação Ambiental ¹).

Na conceção da planificação foi respeitada a Taxonomia de Bloom (atualizada). Esta é uma estrutura organizada hierarquicamente dos objetivos educacionais, que classifica as possibilidades de aprendizagem nos domínios cognitivo, afetivo e psicomotor. Para que os alunos adquiram uma nova habilidade do nível superior, pressupõe-se que tenham adquirido a do nível anterior.

Na programação de computadores a Taxonomia de Bloom (*vide* Figura 3.1), pode ser utilizada para ajudar a identificar as dificuldades dos alunos, como se pode observar os dois primeiros níveis englobam a leitura e compreensão do código, os dois níveis seguintes compreendem a capacidade de escrita de pequenos fragmentos de código, os últimos dois níveis refletem a habilidade de escrita de programas informáticos completos.

¹ Estes domínios são obrigatórios a todos os níveis e ciclos de escolaridade, pois são áreas transversais e longitudinais.

Figura 3.1 - Taxonomia de Bloom - revista (fonte: Anderson & Khathwohl, 2001)



3.3. Conceitos científicos

São vários os conceitos científicos envolvidos na leção das disciplinas de Robótica. Inicialmente estudam-se conceitos de programação, tais como: algoritmos, tipos de dados, variáveis, ciclos, declarações condicionais, lógica aritmética, lógica booleana, listas, entre outros e posteriormente os conceitos associados à robótica como sensores, atuadores, entre outros.

Utilizei alguns destes conceitos, conforme o mapa de conceitos elaborado para a intervenção realizada (*vide* Anexo A).

3.3.1. Ações

As ações ou instruções de ação são basicamente comandos que o computador executa, ou comportamentos que um programa pode realizar.

Estas podem ser úteis na razão que permitem reutilizar o código, como exemplo, nas instruções de ações podemos ter: uma saída de dados para o ecrã; a criação de uma variável; a leitura/escrita de um ficheiro ou o início/paragem de uma execução.

Ação é um acontecimento que, a partir de um estado inicial, após um período de tempo finito, produz um estado final previsível e bem definido (Farrer, 1999).

3.3.2. Algoritmos

Tal como define Martins, (2013) “Um algoritmo é uma sequência finita de instruções bem definidas e não ambíguas, cada uma das quais pode ser executada mecanicamente num período de tempo finito com uma quantidade de esforço finita”.

Existem inúmeros algoritmos que utilizamos no dia-a-dia, como trocar uma lâmpada, fazer uma sandes de queijo, procurar o melhor caminho numa aplicação de mapas ou até para efetuar a programação de um Robot. Estes exemplos mostram como os algoritmos são usados com as sequências finitas para conseguir um determinado objetivo.

Os algoritmos possuem três formas de representação: a narrativa descritiva ou descrição narrativa; o fluxograma convencional e o pseudocódigo também conhecido com linguagem estruturada.

Diz-se que um algoritmo é eficaz quando atinge o objetivo principal, ou seja, a análise da resolução do problema realiza-se prioritariamente.

3.3.3. Estruturas de Controle

Uma estrutura de controle é parte do código que oferece suporte a decisões com base na análise das variáveis, estes componentes funcionais são bastante úteis na programação de computadores. Geralmente estas representam peças funcionais da linguagem de programação e mostram o papel desses processos de tomada de decisão no código e usam condições que são testadas a fim de determinar o fluxo de execução das instruções.

Existem três tipos de estruturas de controle: Estrutura sequencial; Estrutura de decisão e Estrutura de repetição. Hehl (1986) afirma que qualquer algoritmo pode ser escrito combinando-se blocos formados pelas estruturas de controle.

3.3.3.1 Estruturas de Decisão ou Seleção

Na estrutura sequencial, o algoritmo que possui uma estrutura sequencial significa que as suas instruções são executadas na sequência em que elas aparecem, sem nenhuma alteração do seu fluxo, a não ser que, exista alguma instrução explícita para uma mudança deste fluxo.

As estruturas de decisão são aquelas que permitem alterar o fluxo de execução, de forma a selecionar qual a parte do algoritmo que deve ser executada, essa “decisão”

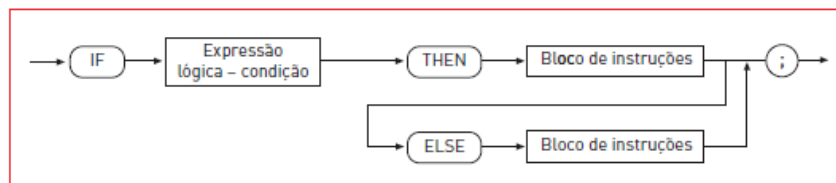
de execução, é tomada a partir de uma condição que pode resultar em Verdadeiro ou Falso, uma condição é ainda representada por expressões relacionais ou lógicas. Por exemplo “*Se hoje não chover, então a Maria irá à praia*”, neste algoritmo podemos ver que a Maria só irá à praia se não chover, ou seja a instrução de “*ir à praia*” só será executada se a condição de “*não chover*” for verdadeira.

Podemos ter três tipos de estrutura de decisão: decisão simples, decisão composta e decisão múltipla.

A estrutura de decisão simples é quando pretendemos que uma determinada instrução ou conjunto de instruções execute apenas se uma determinada condição for verdadeira (e.g. *if* condição *then* instrução). Uma condição deve ter como resultado apenas dois valores possíveis: verdadeiro ou falso. A instrução só será executada se tiver o valor verdadeiro.

```
if (condição) {  
    instrução1;  
    instrução 2;  
    ...  
}
```

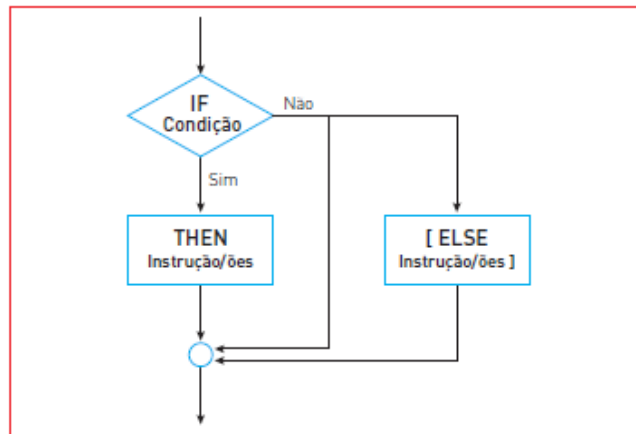
Figura 3.2 – Diagrama sintático da estrutura *if... then... else...* (fonte: Porto Editora, 2022)



A estrutura de decisão composta, é quando temos uma estrutura de decisão e queremos que outra instrução ou um bloco de instruções seja executado, caso a condição de decisão seja falsa (e.g. *if* condição *then* instrução 1 *else* instrução 2). Esta estrutura é chamada de decisão composta.

```
if (condição) {  
    instrução 1;  
} else {  
    instrução 2;  
}
```

Figura 3.3 – Diagrama representativo da instrução ou estrutura *if... then... else...* (fonte: Porto Editora, 2022)

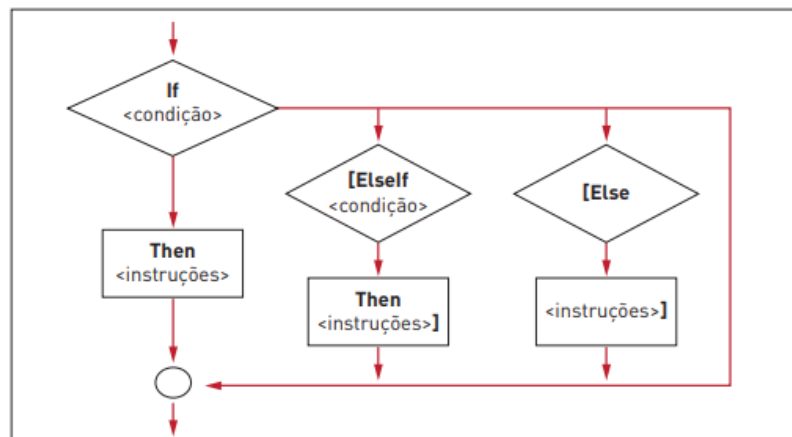


O comando de decisão múltipla é uma outra forma de escrever uma estrutura de condição, como as expressões condicionais são avaliadas de cima para baixo, se uma condição verdadeira é encontrada, o bloco associado a ela será executado e o resto do encadeamento é ignorado (e.g. *if* condição *then* instrução 1 *elseif* instrução 2 *else* instrução 3). Se nenhuma das condições é verdadeira então o *else* final será executado.

Se o *else* final não estiver presente e todas as outras condições forem falsas, então nenhuma ação será realizada.

```
if (condição) {  
    instrução 1;  
} else if (condição) {  
    Instrução 2;  
} else {  
    intrução 3:  
}
```

Figura 3.4 - Representação esquemática - fluxograma de uma estrutura de decisão if... then... else... (fonte: Porto Editora, 2022)



3.3.3.2 Estruturas de Repetição ou Ciclos

As estruturas de repetição são aquelas que permitem executar mais que uma vez uma determinada parte do algoritmo (ciclos), as repetições devem ser sempre finitas.

As instruções de repetição, ou ciclos, permitem a execução de forma repetida de um conjunto de instruções. A execução depende de um valor lógico de uma condição que é testada em cada iteração para decidir se a execução do ciclo continua ou termina (Santos, 2006).

Existem três tipos de ciclos, ciclo condicional: repetir-até (*repeat-until*); ciclo condicional: repetir-enquanto (*do-while*) e o ciclo determinístico faça-para (*do-for*).

O ciclo condicional repetir-até (*repeat-until*): executa um bloco de instruções até que uma determinada condição lógica seja verdadeira. Este ciclo testa a condição lógica após a primeira iteração, ou seja, o teste é realizado à saída. Este ciclo deve ser utilizado sempre que se desejar que o código seja executado pelo menos uma vez.

O ciclo condicional repetir-enquanto (*do-while*): executa um bloco de instruções enquanto uma determinada condição lógica for verdadeira. Este ciclo testa a condição lógica à entrada.

O ciclo determinístico faça-para (*do-for*): executa um bloco de instruções com um número pré-determinado de vezes.

Estas instruções permitem executar mais de uma vez um determinado código, as repetições devem ser sempre finitas e para isso devem utilizar um determinado critério como teste no início, teste no fim ou variável de controle.

3.3.4. Eventos

Em computação, um evento é o resultado de uma ação. A ocorrência de um evento pode provocar uma reação, que pode ser uma ação, ou conjunto de ações a ser tomada. Entende-se como uma mensagem indicando que aconteceu alguma ação, como por exemplo, o pressionar o botão do rato ou de uma tecla no teclado.

Um evento, pode ser quando pressionamos um botão do rato, que determina a ocorrência do clicar do rato (*mouse click*), pode ser a programação um conjunto de instruções para que o programa responda com uma ação a esse evento.

Também podem existir eventos através do teclado, quando pressionamos uma tecla o programa determina a ocorrência da tecla pressionada (*KeyDown*), acompanhado da identificação da tecla.

São ainda considerados eventos, ações como um botão de um joystick, uma entrada de dados no microfone, numa rede, uma entrada numa base de dados, um ficheiro que foi criado, alterado, apagado ou acedido, entre outros.

3.3.5. Programação de Sensores

Um sensor em eletrónica é um dispositivo que capta um fenómeno físico (e.g. temperatura, humidade ou luminosidade) e converte-o em sinal eléctrico. Os sensores fazem parte da interface entre o mundo físico e o mundo dos dispositivos eletrónicos, como os robots e os computadores.

Geralmente, a saída obtida num sensor é de grandeza eléctrica com pequena tensão, ou corrente ou ainda uma alteração num valor da resistência eléctrica, que depois é enviada para um circuito electrónico.

Existem inúmeros formatos de sensores, que possuem determinadas características, e podemos classifica-los quanto a sua categoria, como sensores de controle de: temperatura, luminosidade/óticos, pressão, aceleração, distância/proximidade, sonoro, substâncias, imagem/câmara, magnético, bem como sensores específicos para deteção de outras grandezas físicas.

Quanto a sua aplicabilidade, os sensores estão presentes em quase todas as áreas do conhecimento como na medicina, engenharia, controle de tráfego, segurança, telecomunicações, entre outras. A sua aplicabilidade varia, desde o controle da velocidade de um veículo ou de um airbag, de um sensor de presença, da deteção de intrusos, controle do fluxo de água, abertura e fecho de portas, entre outros.

3.3.6. Operadores Relacionais, Lógicos e Aritméticos

Na escrita dos algoritmos são utilizados operadores relacionais, lógicos e aritméticos, os mais comuns estão de acordo com a semântica definida.

Tabela 3.1- Operadores relacionais, lógicos e aritméticos (fonte: Santos, 2006)

<	menor que			+	soma
>	maior que			-	subtração
≥	maior ou igual que	e, ∧	conjunção	*	multiplicação
≤	menor ou igual que	ou, ∨	disjunção	/	divisão
=	igual	não, ¬	negação	div	divisão inteira
≠	diferente			%	resto da divisão inteira

Os operadores aritméticos tomam operandos numéricos (dos tipos inteiro ou real) e produzem resultados também numéricos.

Os operadores relacionais tomam operandos numéricos (dos tipos inteiro ou real) e produzem resultados booleanos (verdadeiro ou falso).

Os operadores lógicos tomam operandos booleanos e produzem resultados booleanos.

3.3.7. Tipos de Dados

Uma das principais funcionalidades de um computador é a manipulação de informações, assim, é necessário que existam diversas formas de trabalhar com diferentes tipos de dados num programa. Apesar de internamente o computador manipular unicamente números, as linguagens de programação permitem que se utilize outros tipos de informação nos programas de forma transparente.

Na programação de computadores um dado é algo que pode ser representado, armazenado e manipulado, desta forma, podemos também definir um tipo de dados como um conjunto de objetos que tem em comum o mesmo comportamento diante de um conjunto definido de operações. Alguns dos tipos de dados definidos na maioria das linguagens de programação são os números inteiros, números reais, letras e objetos lógicos. Dentre estes, talvez o tipo menos conhecido seja o dos valores lógicos, também conhecido como booleanos (Medina & Fertig, 2005).

As linguagens de programação implementam formas de representar e manipular esses dados que podem ser classificados em dois grandes grupos: os tipos de dados que são primitivos e os tipos de dados que não são primitivos.

Os tipos de dados primitivos são os tipos básicos que devem ser implementados por todas as linguagens de programação, como os números reais, inteiros, lógicos, caracteres e uma cadeia de caracteres (*string*). Os tipos de dados não primitivos, normalmente são os vetores, matrizes, classes, que são estruturas de dados mais complexas do que os tipos de dados primitivos.

3.3.8. Variáveis

As variáveis são um recurso que os programas usam para ler e escrever dados na memória de um computador, é um espaço na memória que o computador reserva e atribui-lhe um determinado nome.

Embora uma variável possa assumir diferentes valores, ela só pode armazenar um único valor a cada instante. Toda a variável é identificada por um nome ou identificador. Quando declaramos uma variável, significa dizer que criamos (definimos) locais na memória do computador rotulados com os nomes das variáveis (identificadores) (Baudson & Araújo, 2013).

Também é possível de definir o tipo de dados que estas podem conter, ou seja, as variáveis podem conter dados do tipo inteiros ou reais (números), caracteres (letras, símbolos) ou lógico (V/F).

As variáveis são importantes para os programas que contêm cálculos, condições, repetições ou outros dados mutáveis durante a sua execução.

4. Problematização da Temática a Lecionar

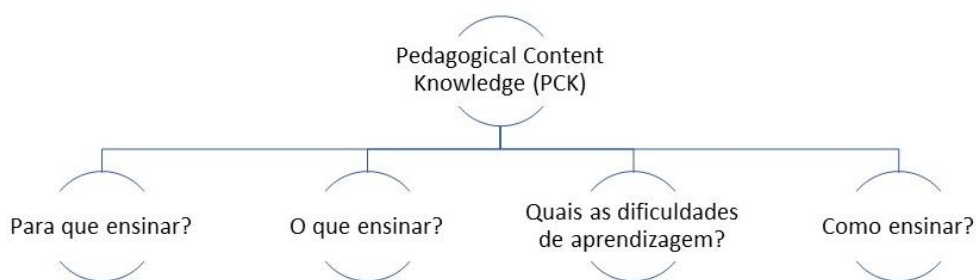
O ensino da programação tem como principal objetivo o desenvolvimento nos alunos de capacidades, conhecimentos e competências para colocar num programa de computador a resolução dos problemas do dia-a-dia.

Durante o processo de ensino-aprendizagem de fundamentos de programação nota-se que grande parte dos alunos apresenta dificuldades em assimilar as abstrações envolvidas (Junior & Rapkiewicz, 2005).

A programação é só um dos tópicos de preocupação no ensino da informática (Saeli et al., 2011). Também Kurland et al., (1989), (citado por Saeli et al., 2011) indica que a programação é uma aptidão que é difícil de aprender mesmo ao fim de dois anos de instrução.

Shulman, (1987), (citado por Saeli et al., 2011), define o “*Pedagogical Content Knowledge*” (PCK), como “As formas de representação e formulação do tema que o tornam compreensível para os outros”. De facto, existe uma diferença entre saber programar e ser capaz de ensinar programação. A sala de aula, onde a aprendizagem e o ensino ocorrem, é um ambiente complexo em que vários processos e ações acontecem. Mas quando se fala de PCK deve ser ter uma atenção especial com a aprendizagem dos alunos. Um aspeto do PCK diz respeito aos professores, estes têm de saber formular e apresentar a matéria, para que a compreensão possa ocorrer, dado que sabemos que os alunos têm diferentes estilos e necessidades de aprendizagem.

Figura 4.1 - Fases do Pedagogical Content Knowledge (fonte: Shulman, 1987)



A questão usada para desvendar o PCK é: como deve o tópico ser ensinado? Ao responder a estas quatro perguntas (*vide* Figura 4.1) será possível definir o PCK de um determinado assunto. As quatro perguntas estão todas ligadas entre si, porque as razões para ensinar um tópico (primeira pergunta) influenciarão os conteúdos escolhidos para serem incluídos no currículo. Além disso, as dificuldades de

aprendizagem que os estudantes encontram certamente influenciarão a forma de o ensinar.

Para o ensino da programação no caso desta intervenção foi necessário pensar em três temáticas fundamentais: O Pensamento Computacional (PC); As Dificuldades no Ensino da Programação (DEP) e por último a Robótica Educativa (RE) todas como estratégias pedagógicas para uma aprendizagem efetiva.

4.1. Pensamento Computacional

O Pensamento Computacional, é um tema bastante importante que tem recebido uma enorme atenção por parte dos investigadores em educação. Wing (2006) define-o como uma forma que envolve a resolução de problemas, projetar os sistemas e compreender os comportamentos dos seres humanos, baseando-se nos conceitos fundamentais da ciência da computação, também afirma que este usa a abstração e decomposição para atacar uma tarefa grande e complexa ou projetar um sistema complexo. Mais recentemente a mesma autora (Wing, 2011), atualizou a definição do Pensamento Computacional e propôs que ainda se incluísse o processo da formulação de problemas e das suas soluções que sejam representadas de forma que possam ser efetivamente realizadas por um agente de processamento de informações.

Mais recentemente Brennan e Resnick (2012) apontam ainda ao Pensamento Computacional outras competências relacionadas com a resolução de problemas, a compreensão dos problemas, a definição de problemas, a abstração, o pensamento lógico, a depuração e o reconhecimento de padrões.

Existem ainda várias iniciativas interessantes para a promoção desta temática, como o desenvolvimento de competições sobre o Pensamento Computacional (*vide* www.bebras.uk e bebras.dcc.fc.up.pt), esta iniciativa internacional que se realiza anualmente, foi criada para promover e introduzir a informática e o Pensamento Computacional para estudantes de todas as idades, com ou sem experiência prévia.

O Pensamento Computacional que foi apontado pela Organização para a Cooperação e Desenvolvimento Económico (OCDE) como um dos elementos-chave a integrar nos currículos do século XXI.

4.2. As Principais Dificuldades no Ensino da Programação

Uma questão não menos importante é a Dificuldade no Ensino da Programação, tal como refere Jenkins (2002) a aprendizagem da programação de computadores é um desafio e um processo complexo para muitos estudantes.

Para Gomes et al., (2008) os elevados níveis de insucesso em disciplinas onde são ensinados os conceitos mais básicos de programação, em qualquer grau e sistema de ensino, é um problema universal que tem sido alvo de variadas pesquisas, resultando também em diversificados sistemas, sem que contudo o panorama tenha melhorado significativamente. Existe um conjunto de razões que estão na origem do problema, nomeadamente, métodos de ensino e aprendizagem desadequados, falta de vários tipos de competências por parte dos alunos, em particular no que respeita à resolução de problemas, a difícil natureza do tema e uma forte conotação negativa que lhe está associada.

As metodologias tradicionalmente utilizadas para aprender/ensinar estes assuntos não se revelam suficientes, pelo que é apresentada uma nova proposta, centrada na atividade de resolução de problemas, de acordo com o nível cognitivo do aluno e os seus estilos preferenciais de aprendizagem (Piedade, 2021).

4.2.1. A Complexidade da Programação

No seu nível mais elevado de proficiência a programação é muito mais do que a escrita de um conjunto de linhas de código numa dada linguagem, é uma arte e uma ciência. Arte porque existem muitas maneiras diferentes de codificar instruções, com alguma criatividade. É também uma ciência, porque é constituída por um conjunto de regras orientadoras, porque é necessário o uso de lógica e porque existem alguns métodos rigorosos de programação que asseguram a eficiência, economia e utilidade dos programas gerados (Gomes et al., 2008).

Para Saeli et al., (2011) um dos principais aspetos na complexidade da programação é a dificuldade em compreender a utilidade da programação e aplicar corretamente a sintaxe das linguagens de programação.

Também Piedade et al., (2019) apontam que são necessárias tanto a compreensão teórica como a aplicação prática da semântica e da sintaxe específicas das linguagens de programação para a sua compreensão.

Vários autores consideram que a programação é uma disciplina muito exigente e difícil de dominar, associando essa dificuldade à sua natureza abstrata, assim a experiência tem demonstrado que em termos genéricos, existir uma grande dificuldade em compreender e aplicar certos conceitos abstratos de programação.

Uma grande dificuldade resulta do facto dos alunos não conseguirem compreender e aplicar noções básicas de programação de modo a resolverem problemas concretos, estas dificuldades surgem independentemente do tipo de linguagem de programação usada.

Numa tentativa de superar estas dificuldades, alguns estudos apontam a Robótica Pedagógica como um bom aliado na resolução de alguns dos problemas identificados (Fernandes, 2013).

4.3. Disciplina de Introdução à Robótica

Tal como já referido anteriormente esta disciplina foi idealizada para ter enfoque na programação de computadores através da robótica, procurando assim práticas diferenciadas em áreas que são cada vez mais procuradas no mercado de trabalho.

Os robots utilizados foram os *mBot*, estes utilizam uma Linguagem Visual de Programação por Blocos (LVPB) baseada em *Scratch 3*, esta plataforma chamada de *mBlock* possui uma versão online (<https://ide.mblock.cc/>) e ainda as versões de instalação física para os diferentes sistemas operativos.

A utilização das linguagens de programação por blocos em detrimento das linguagens textuais, permite ao utilizador final um ambiente mais amigável e assim possibilita uma melhor aprendizagem das linguagens de programação, (e.g. não tem as sintaxes complexas presentes nas linguagens de programação textuais).

“The visual programming environment (VPE) and block-based programming environment have been referred in many research studies as important tools to learn programming and to solve many problems taught in primary schools. In fact, visual programming with blocks is a great support in introducing programming and saves pupils from the difficulties of traditionally complex text-based languages” (Piedade et al., 2019. p. 4).

Piedade et al., (2019)), referem que o estudo e programação dos objetos tangíveis e a robótica por parte dos estudantes, é um caminho pedagógico para o desenvolvimento do pensamento computacional e para o desenvolvimento das competências na programação.

Além de introduzir aos alunos os conceitos fundamentais de programação, procurou-se nesta disciplina também introduzir os alunos aos conceitos iniciais de robótica, com atividades individuais, em pequeno e grande grupo por forma a estimular a criatividade, o pensamento crítico, o trabalho cooperativo e colaborativo através de atividades experimentais

Piedade et al. (2019) referem que as atividades educacionais baseadas no uso da robótica podem ajudar os alunos a assumir um papel mais ativo no seu processo de aprendizagem, e assim desenvolver diversas competências mentais e criar novo conhecimento.

4.3.1. A Robótica Educativa

A Robótica Didática (ou pedagógica ou educacional) são termos utilizados para caracterizar ambientes de aprendizagem que reúnem plataformas robóticas ou kits de montagem compostos por peças diversas, atuadores e sensores, controláveis por micro controladores e/ou computadores, com softwares que permitam programar de alguma forma o funcionamento dos modelos montados (Oliveira, 2008).

A revisão da literatura revela que no ensino básico a robótica educativa tem sido utilizada como ferramenta de ensino e aprendizagem de várias disciplinas, com especial ênfase em projetos interdisciplinares envolvendo a Matemática, as Ciências, e em especial a Informática e a Tecnologia.

Ferreira et al., (2010), consideram que hoje em dia é evidente que o ensino não pode ser imutável, a introdução de novas tecnologias torna-se natural e pode ser usada como um instrumento de combate ao abandono escolar, destacando a motivação para o uso da robótica como um dos instrumentos didáticos mais poderosos entre as atuais novas tecnologias.

A robótica permite tornar tangíveis os conceitos ligados à programação e ao pensamento computacional, ou seja, fora do espaço de um ecrã de computador. Deste modo, é possível aprender a criar, planejar, resolver problemas, programando artefactos

tangíveis, e conseqüentemente, construindo algo com uma finalidade que proporcione a articulação com outros conteúdos das diferentes áreas do saber.

A robótica propicia uma aprendizagem mais profunda da tecnologia, promovendo momentos para “aprender fazendo”, de forma tátil, na relação que o aluno estabelece ao relacionar as suas ideias com os artefactos, processo durante o qual os alunos obtêm e visualizam resultados imediatos (Pedro et al., 2017).

Chalmers (2018), indica que a Robótica Educativa pode ser um caminho seguro para introduzir o Pensamento Computacional e a programação desde os primeiros anos de escolaridade através de atividades que envolvam os alunos a implementar a sequência passo-a-passo do código do programa a fim de permitir que o robot resolva o problema ou pelo menos parte dele. De acordo com Bers et al. (2014), os jovens estudantes podem simplesmente usar linguagens de programação baseadas em blocos e a robótica como forma de desenvolver as suas competências de código e o pensamento computacional.

A integração da robótica em contexto educativo permite criar cenários de aprendizagem diversificados, que reúnem tecnologia, linguagens de programação e objetos tangíveis; promovendo-se assim a articulação com as áreas curriculares e/ou transversais, onde se realizam projetos contextualizados que no seu conjunto proporcionam aos alunos a oportunidade de desenvolver a sua criatividade e ter um papel ativo na construção do seu próprio conhecimento (Pedro et al., 2017).

5. Intervenção Pedagógica

A Prática de Ensino Supervisionada, foi realizada no âmbito da disciplina de Introdução à Robótica, especificamente no módulo 3, na turma C do 9.º ano de escolaridade do Agrupamento de Escolas de Portela e Moscavide. Assim apresento os vários documentos utilizados como suporte à intervenção realizada, como os respetivos planos de aula o cenário de aprendizagem, os objetivos, os recursos, a metodologia utilizada e estratégias adotadas.

5.1. Observação de aulas

A observação desempenha um papel fundamental na melhoria da qualidade do ensino e da aprendizagem, constituindo uma fonte de inspiração e motivação e um forte catalisador de mudança na escola (Reis, 2011).

Antecedendo a intervenção, iniciei a minha atividade na Escola Secundária da Portela, com a observação de 6 aulas do Professor Cooperante no 1.º e 2.º períodos letivos das 11 previstas, as 5 aulas que não foram observadas deveram-se à participação do professor cooperante no projeto *Erasmus+* que o obrigou a ausentar-se do país, bem como na receção de grupos de professores e alunos estrangeiros. Assim, elaborei grelhas de observação tanto para a observação do Professor Cooperante (*vide* Anexo B) como para a observação dos alunos da turma (*vide* Anexo C).

No dia 13 de outubro assisti a primeira aula do professor cooperante, na qual ele fez a minha apresentação e explicou o motivo da minha presença, após isso, indicou a tarefa que os alunos deveriam realizar no decurso da aula, que foi a participação num concurso de programação numa plataforma online. O professor teve o cuidado de deixar trabalho atribuído dado que iria estar ausente nos 15 dias seguintes devido ao projeto Erasmus+.

No dia 3 de novembro, segunda aula observada, o professor cooperante esteve a verificar o trabalho realizado pelos alunos durante a sua ausência, dando um feedback formativo sobre o desempenho dos alunos na atividade proposta.

No dia 17 de novembro, terceira aula observada, o professor cooperante dividiu a turma em grupos e lecionou uma aula prática de montagem de robots e os seus respetivos sensores/atuadores, os grupos participaram procedendo a assemblagem dos

robots do grupo. Durante esta aula foi possível perceber que o docente além de aferir se o material estaria todo em boas condições de funcionamento, também aproveitou para introduzir alguns conceitos básicos de eletrônica e mecânica presentes nos robots utilizados.

No dia 15 de dezembro, após 3 semanas sem aulas (1 aula devido à recepção de um grupo de professores e alunos estrangeiros e 2 aulas devido aos feriados nacionais), o professor continuou a verificar e testar o material eletrônico e a proceder à substituição de componentes dos robots que não estavam nas melhores condições, e fazer testes práticos de verificação de velocidade, potência, rotação, entre outros.

No dia 19 de janeiro (dia 12 não houve aula devido a uma recepção a professores do projeto *Erasmus+*), o professor procedeu a uma introdução à linguagem de programação por blocos através da plataforma *mBlock*, na qual esteve a explicar as suas características (área de trabalho, os blocos de programação, a área de programação por código, a importância do *login*, a componente do armazenamento e a partilha dos projetos na nuvem (*cloud*), foram ainda realizados alguns programas simples para demonstrar o funcionamento dos blocos de programação.

No dia 26 de janeiro o professor continuou a demonstração da plataforma *mBlock*, a instalação de dispositivos (na biblioteca da plataforma) e os seus blocos de programação, bem como as extensões adicionais dos dispositivos que permitem aumentar exponencialmente a capacidade de programação dos robots com o uso de sensores, inteligência artificial, internet das coisas, entre outros. Ainda conseguiu demonstrar como se carrega o programa do *mBlock* para o robot *mBot*.

Assim, através dos documentos de trabalho desenvolvidos, fui tirando notas de campo e efetuando registos escritos que foram importantes para perceber e compreender as técnicas, estratégias e metodologias de ensino utilizadas pelo professor cooperante e também para entender as dinâmicas em sala de aula, tanto no trabalho individual como no trabalho em grupo dos alunos.

Quanto a observação do professor cooperante, verifiquei que este possui uma enorme experiência no que diz respeito à utilização da robótica educativa nas suas atividades letivas, tanto a nível nacional como internacional, ele expressou-se bem e forneceu instruções de forma clara e concisa, conseguiu utilizar efetivamente os conhecimentos prévio dos alunos e diversificou as atividades em sala de aula. Reparei ainda que ele revelou um entusiasmo puro, sincero e contagiante nas atividades letivas e adequou as estratégias de ensino aos conteúdos a lecionar, conseguiu captar a atenção

dos alunos e estimulou e encorajou a participação destes, resumindo o professor cooperante demonstrou excelentes conhecimentos dos conteúdos e proporcionou momentos de aprendizagem efetivos e estimulantes em sala de aula.

Na observação dos alunos devido a todas as condicionantes existentes (as aulas que não se realizaram e os casos de isolamento devido à Covid19), foi só possível de observar um determinado grupo de raparigas e rapazes (3 elementos de cada género), mas como resultado dessa observação foi possível aferir que existiu uma grande receptividade pelos alunos da disciplina e principalmente dos conteúdos que foram lecionados, observei ainda que os alunos envolveram-se com entusiasmo e apropriaram-se das aprendizagens. O discurso dos alunos foi francamente positivo relativamente às aprendizagens realizadas e o ambiente em sala de aula foi claramente de participação, interesse e motivação.

Observei e discuti com o professor cooperante as características da maioria dos alunos da turma, tendo denotado ainda algumas dificuldades presentes nos alunos, sendo que as mais relevantes prenderam-se com as dificuldades na utilização da programação, principalmente pelos alunos que estiveram em contacto com a disciplina pela primeira vez.

Desta forma estou certo que a observação realizada foi importante para a planificação adequada e para a preparação das estratégias de ensino, procurei assim adaptar as atividades de acordo com as características dos alunos, privilegiando uma metodologia de cariz prático.

5.2. Cenário de Aprendizagem

Um cenário de aprendizagem é uma ferramenta que tem um papel estratégico na Formação Inicial de Professores (FIP).

Este conceito tem sido utilizado em diferentes áreas de atividade como estratégia para projetar o futuro, antecipando problemas e prevendo soluções para esses mesmos problemas (Matos et al., 2019).

“A utilização de cenários em situações de educação e formação apela ao desenvolvimento da capacidade de imaginar de forma fundamentada e sustentada situações educativas com objetivos claros e específico.

Identifica-se por isso uma relação natural entre essa capacidade e competências relacionadas com o pensamento crítico e criativo, a resolução de problemas, a

comunicação e a colaboração, competências estas que têm vindo a ser recorrentemente referidas em vários referenciais internacionais” (Matos et al., 2019, pp. 227-228).

Figura 5.1 - Princípios orientadores para o desenho de cenários de aprendizagem (fonte: Matos, 2014)



Matos et al., (2019), refere ainda que a conceção de um cenário de aprendizagem é algo que o professor faz regularmente, sempre que planeia atividades de ensino, e procurar antecipar ou desenhar experiências e problemas que pretende fazer acontecer na sala de aula com os seus alunos (pp. 228).

Assim, desenvolvi um cenário de aprendizagem (*vide* Anexo D), no qual pretendi na sua génese que os alunos através da Introdução à Robótica trabalhassem a aprendizagem das linguagens de programação e, através da resolução de problemas articulassem os conhecimentos adquiridos nas diferentes disciplinas desenvolvendo assim, a capacidade de pensar os problemas que enfrentamos com o crescente crescimento urbanístico e populacional nas cidades, por último, que pensassem como poderiam contribuir para as soluções que precunizam as “*Smart Cities*”.

Os problemas foram apresentados e resolvidos em cada aula, pelos respetivos grupos, apresentando as suas soluções e submetendo-as na plataforma da turma para posterior análise.

Os alunos foram organizados em grupos de 4 elementos, sendo 2 grupos de raparigas e 4 grupos de rapazes, que assim contribuíram para a investigação realizada nesta intervenção.

5.3. Objetivos de Aprendizagem

Os objetivos de aprendizagem que procurei desenvolver com os alunos na intervenção foram os seguintes:

- Relacionar a programação e a robótica na resolução de problemas;
- Reconhecer os princípios e conceitos fundamentais da programação;
- Reconhecer as vantagens do pensamento computacional;
- Reconhecer a vantagem da reutilização do código;
- Reconhecer a vantagem da programação de robots para a aprendizagem das linguagens de programação;
- Identificar um problema e decompô-lo em subprogramas;
- Identificar componentes estruturais de programação;
- Identificar os diferentes tipos de dados;
- Identificar os diferentes operadores e funções;
- Demonstrar a aprendizagem da programação;
- Experimentar o funcionamento das estruturas de seleção e controlo;
- Integrar as variáveis e sequências para a sua utilização;
- Resolver erros existentes na programação e na sequência da resolução do problema;
- Elaborar sequências de instruções para utilização das estruturas de seleção e controlo.

5.4. Metodologia de avaliação das aprendizagens

A avaliação deve ser uma ação contínua e sistemática integrada no processo de ensino-aprendizagem e não desempenhar apenas uma função classificadora dos conhecimentos dos alunos num determinado momento (Dorotea, 2013).

Nesta intervenção propus-me a fazer três tipos de avaliação, uma avaliação diagnóstica antes do início da intervenção, uma avaliação formativa durante a intervenção e uma avaliação sumativa no final da intervenção

A avaliação formativa desempenha um papel fundamental no processo de ensino e de aprendizagem; informa sobre o estado dos alunos, os problemas que apresentam, como estão a progredir. Um *feedback* relevante e imediato promove a autonomia dos alunos e a autorregulação dos seus próprios

processos de aprendizagem, e permite ao professor decidir sobre o processo de ensino, promovendo a adaptação de estratégias de acordo com as necessidades dos alunos (Dorotea & Pedro, 2015).

Antes da intervenção (início de fevereiro) foi aplicada a avaliação diagnóstica, aferindo assim os conhecimentos prévios dos alunos, esta contribuiu para a preparação e operacionalização do cenário e elaboração dos materiais de acordo com a necessidade específicas de alguns alunos.

A avaliação formativa foi efetuada continuamente, através da grelha de verificação de observação direta (com o recurso a rubricas de avaliação formativa) (*vide* Anexo W), regista-se o interesse, a participação nas tarefas, a qualidade de trabalho realizado, a gestão das tarefas e a capacidade de organização.

Assim obtive uma informação que me permitiu orientar o processo de desenvolvimento das aprendizagens, e fosse adequando-o às características dos alunos, permitiu também a adaptação das atividades letivas às dificuldades individuais de cada um.

Esta avaliação deve ser simples e clara para que seja compreendida pelos alunos, e lhes permita regular a sua aprendizagem, por forma a permitir que cada um construa o seu percurso de aprendizagem.

Uma definição da avaliação sumativa é dada pelo Decreto-Lei n.º 139/2012, de 5 de julho, que no seu artigo n.º 24 diz que: “A avaliação sumativa traduz o juízo global sobre as aprendizagens feitas pelos alunos, tendo como objetivo a classificação e certificação”.

A avaliação sumativa (com o recurso a rubricas de avaliação) (*vide* Anexo X) foi efetuada após o trabalho individual e do grupo na resolução dos problemas apresentados, avaliei a conceção, a realização e apresentação da solução para cada problema, pretendi que esta avaliação refletisse a aprendizagem efetuada ao longo desta intervenção e assim fosse traduzida posteriormente numa classificação.

O processo avaliativo é um processo de grande importância pelo impacto que este causa na vida académica dos alunos, assim este deve ser encarado com responsabilidade e proporcionar aos alunos atingir os seus objetivos de aquisição dos conhecimentos, e não ser só, um processo de mera classificação.

Quando a avaliação é entendida apenas como mera forma de reduzir toda a informação a uma classificação, transforma-se num instrumento de premiação ou punição dos alunos (aprovação ou reprovação) (Dorotea & Pedro, 2015).

5.5. Metodologia de Aprendizagem Baseadas em Problemas

Esta metodologia de aprendizagem tem como principal pressuposto que os alunos explorem os seus conhecimentos para tentar resolver um dado problema.

Quando identificarem os conhecimentos necessários para aprenderem a resolver o problema, os alunos investigam sobre o problema e procuram soluções. Caso sejam encontradas várias possibilidades de resolução do problema, estas devem ser avaliadas por forma a que se escolha uma e finalmente proceder à solução do problema. Por fim esta metodologia termina com a apresentação dos resultados alcançados, aqui o aluno é responsável pela sua própria aprendizagem.

Figura 5.2 - Esquema da Aprendizagem Baseada em Problemas (fonte: Piedade et al., 2017)



Pedro et al., (2017), afirmam que esta metodologia promove o trabalho em equipa, a colaboração, o pensamento crítico e criativo, a resolução de problemas, o desenvolvimento de competências de comunicação, entre outras competências para o século XXI.

Na aprendizagem baseada em problemas o resultado final pode ser apenas uma ou várias propostas de solução para o problema (Pedro et al., 2017).

5.6. Atividades

As atividades realizadas na minha intervenção foram as seguintes:

- Avaliação diagnóstica inicial;
- Apresentação da intervenção aos alunos;
- Programação de robots sem sensores;
- Programação de robots com sensor de ultrassom e sensor seguidor de linha (com e sem obstáculos);
- Programação de robots com o módulo *LED RGB e o Buzzer*;

- Programação de robots com o sensor de luz e o módulo LED RGB;
- Programação de robots com sensor de movimento PIR (Passive Infrared Sensor);
- Avaliação Formativa;
- Avaliação Sumativa.

5.7. Recursos

Os Recursos Educativos Digitais (RED) que foram colocados a disposição dos alunos, foram fundamentais para a execução das estratégias pedagógicas, e assim, suportaram a prática de ensino supervisionada.

Estes recursos podem ser facilmente atualizados, podem ser incluídos em redes de colaboração e comunicação, proporcionando a aprendizagem individual ou em grupo, estimulando a aprendizagem criativa

Estes foram também importantes no apoio aos alunos no trabalho síncrono e assíncrono, bem como, para a consolidação das suas aprendizagens em qualquer hora e em qualquer lugar.

Revestiu-se de especial importância a planificação e preparação destes de forma que fossem atuais, úteis, intuitivos, objetivos, e possuíssem rigor científico.

Para tal mobilizei os seguintes recursos:

- Computadores com ligação à Internet;
- *Smartboard Promethean Interactive WhiteBoard*;
- Kit Robot Educativo *Makeblock mBot v1*;
- Quadros brancos e canetas;
- Planos de aula;
- Apresentações eletrónicas;
- Guiões de trabalho;
- Problemas da aula;
- Ficha de avaliação (inicial/final);
- Grelhas de avaliação (formativa/sumativa);
- Software de apoio (*Google Classroom, Google Docs, mBlock 5*).

5.8. Operacionalização das atividades letivas

Para a operacionalização das atividades letivas foram criados os respetivos planos de aula (*vide* Anexos E a I) com a finalidade de definir os seguintes elementos: os objetivos específicos, os conteúdos, as atividades, as metodologias/estratégias, os recursos técnicos/pedagógicos e os critérios e instrumentos de avaliação a adotar em cada aula.

Entendo que o meu papel, como Professor Estagiário (PE), foi o de proporcionar aos alunos autonomia na construção do conhecimento, no desenvolvimento do raciocínio lógico, na capacidade da análise crítica, tanto no trabalho individual como colaborativo. A minha função foi procurar ser um orientador no pensamento e na pesquisa em busca do conhecimento para a resolução dos problemas propostos nas atividades letivas e, permitir ao aluno ser o protagonista da sua aprendizagem envolvido na situação-problema.

Assim, para a concretização das atividades letivas, concebi uma disciplina na Plataforma Google Classroom (PGC) adotada pela escola (*vide* Anexo J) na qual estruturei toda a intervenção. Esta continha uma secção para apoio onde constavam os documentos orientadores (estrutura modular da disciplina, critérios de avaliação e calendarização da prática de ensino supervisionada), as secções para as aulas lecionadas (5 sessões de 100 minutos), uma secção com os documentos de avaliação (rubricas de avaliação formativa e grelhas de avaliação sumativas) e, na última secção, os recursos para apoio aos alunos, em específico, apresentações eletrónicas (*vide* Anexo V), manuais, guiões (*vide* Anexo Z a VV) e vídeos de apoio.

Figura 5.3 - Posto de trabalho dos grupos



Em cada problema proposto, os alunos apoiavam-se nas aprendizagens anteriores e em grupo procuravam superar as dificuldades encontradas, e melhorar sempre o código com a finalidade de levar o *robot* executar a tarefa sem erros. Quando surgiam erros ou dificuldades que não conseguiam superar aí solicitavam a ajuda do Professor Estagiário.

Os problemas idealizados tiveram em conta os problemas do dia-a-dia, estes foram pensados com o intuito de proporcionar e promover o trabalho colaborativo entre alunos, sendo que na fase da conceção dos mesmos procurei auscultar o Professor Cooperante para obter a sua opinião técnica e científica e grau de adequação aos alunos em causa.

Ao longo desta intervenção procurei aplicar a metodologia de Aprendizagem Baseada em Problemas (APB), aumentando gradualmente a dificuldade ao longo da intervenção, procurei sempre colocar exemplos de código que demonstrassem o que era pretendido e programei os *robots* para demonstrar a solução e assim, os alunos perceberem claramente o que era esperado que fizessem.

Tive sempre o cuidado de questionar e desafiar os alunos para compreenderem e melhorarem a programação do robot, procurando eliminar as redundâncias e simplificar o código com o recurso a blocos que otimizassem a programação.

Além da disponibilização dos materiais na plataforma da disciplina, esta ferramenta serviu para a comunicação com a turma e para individualmente dar *feedback* aos trabalhos apresentados, prestar apoio quando os alunos assim o solicitavam e ainda para os alunos submeterem os ficheiros de código para avaliação.

Foram constituídos 6 grupos de trabalho com 4 alunos (sendo que 1 grupo tinha 5 alunos). Cada grupo tinha acesso a um 1 *kit mBot* e um computador portátil, tendo assim que trabalhar colaborativamente na resolução dos problemas propostos.

Na planificação inicial da Prática de Ensino Supervisionada, as atividades letivas estavam previstas terem a duração de 50 minutos e decorrer no período de 23 de fevereiro a 27 de abril (conforme o horário da disciplina). No entanto, foi possível através de uma permuta com a professora que lecionava a aula anterior passar a usufruir de aulas de 100 minutos, tendo a intervenção ficado prevista decorrer assim entre 23 de fevereiro e 23 de março. Contudo, devido às deslocações do professor cooperante no decurso do projeto *Erasmus+* em que participou e a receção dos professores estrangeiros integrantes do mesmo projeto, a necessidade da professora permutadora de realizar momentos de avaliação, e ainda, ao período da interrupção

letiva do 2.º período, a intervenção acabou se prolongar no tempo, decorrendo entre 23 de fevereiro e 4 de maio.

5.8.1. 1.ª e 2.ª aula - 23.02.2022

Na primeira e segunda aula procedi ao registo de presenças após o qual realizei a apresentação da intervenção e do cenário de aprendizagem, em concreto os objetivos a atingir por parte dos alunos, os critérios de avaliação da intervenção e a constituição dos grupos de trabalho.

Figura 5.4 - Grupos de trabalho a programar Robots



De seguida através de uma apresentação eletrónica (*vide* Anexo K) iniciei com uma revisão de conceitos genéricos sobre algoritmia, robótica e da programação por blocos. Quanto à introdução dos novos conceitos, foi apresentado um problema que os alunos resolveram (sem a utilização de sensores), tiveram de perceber qual seria a ação do robot, fazer a simulação do problema, pensar na melhor solução, testar o comportamento do robot e se necessário retificar e melhorar a programação deste.

Por último, através da plataforma da disciplina, foram submetidas as soluções do problema realizadas pelos diferentes grupos para posterior avaliação.

A aula terminou com uma breve síntese dos conteúdos lecionados.

5.8.2. 3.^a e 4.^a aula - 02.03.2022

Na terceira e quarta aula, iniciei as atividades procedendo ao registo de presenças. Foi necessário voltar a rever a apresentação eletrónica da aula anterior sobre o “conceito de algoritmo”, dado que existiram alunos que após serem questionados, demonstraram dúvidas quanto a esses conteúdos. De seguida, procedi à introdução dos novos conteúdos, utilizando uma apresentação eletrónica (*vide* Anexo L) sobre o funcionamento do sensor ultrassónico e o sensor seguidor de linha, bem como os procedimentos para a sua colocação e configuração (ligações às respetivas portas na placa *Arduino*) no robot *mBot*.

Foi apresentado um “cenário da cidade” com a simulação de veículos avariados na via pública, para qual os robots utilizando o sensor ultrassónico, deviam parar quando detetassem os obstáculos, quando finalizassem essa tarefa deveriam programar e testar o sensor seguidor de linha no percurso previamente assinalado.

Figura 5.5 - Grupo de raparigas a programar o Robot



Assim, os alunos idealizaram a ação do robot, com a utilização dos sensores de ultrassom e de seguidor de linha, retificaram e melhoraram a programação deste.

Por último submeteram as soluções a que chegaram na plataforma da disciplina para posterior avaliação.

A aula terminou com uma breve síntese dos conteúdos lecionados.

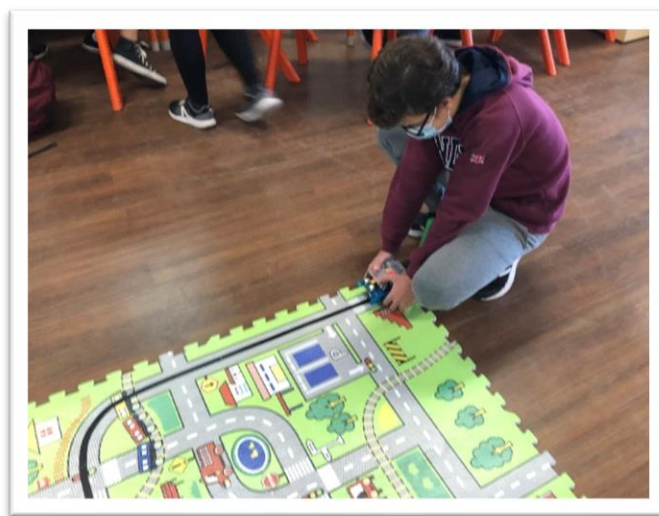
5.8.3. 5.^a e 6.^a aula - 16.03.2022

Na quinta e sexta aula, procedi ao registo de presenças, e efetuei uma revisão da aula anterior (verifiquei nos problemas submetidos na plataforma que continham erros na programação), principalmente o código relativo ao sensor seguidor de linha, ou seja, o comportamento do sensor durante a receção do sinal dos infravermelhos e quando tem de corrigir a trajetória para ficar dentro da linha.

Inicialmente estava previsto serem apresentados nesta aula o sensor de som e o módulo RGB, mas infelizmente não foi possível obter o sensor de som (nem por aquisição, nem por empréstimo), assim foi necessário repensar a planificação inicial e consultando o Professor Cooperante resolvi alterar os conteúdos a lecionar nestas aulas. Utilizei uma apresentação eletrónica (*vide* Anexo M) com a introdução dos novos conceitos e também a associação das funcionalidades tecnológicas (sensores e atuadores) do robot *mBot*.

Nestas aulas os alunos idealizaram a ação do robot, com a utilização dos *LED's* e o *Buzzer* incorporados na placa *Arduino* do *robot*. Para tal verificaram como é constituída a paleta de cores *RGB* (*Red, Green, Blue*) e os respetivos blocos de programação, tal como, o funcionamento do *Buzzer* para reproduzir sons e os respetivos blocos de programação de ambos, como exemplo, foi efetuada a simulação de uma viatura de emergência, que numa determinada situação tinha de ligar os *LED's* de aviso e a respetiva sirene (sinalização de emergência luminosa e sonora).

Figura 5.6 - Aluno a testar Robot no cenário



Assim os alunos programaram, identificaram e retificaram os erros melhorando a programação do robot, e no final submeteram as soluções na plataforma da disciplina.

A aula terminou com uma breve síntese dos conteúdos lecionados.

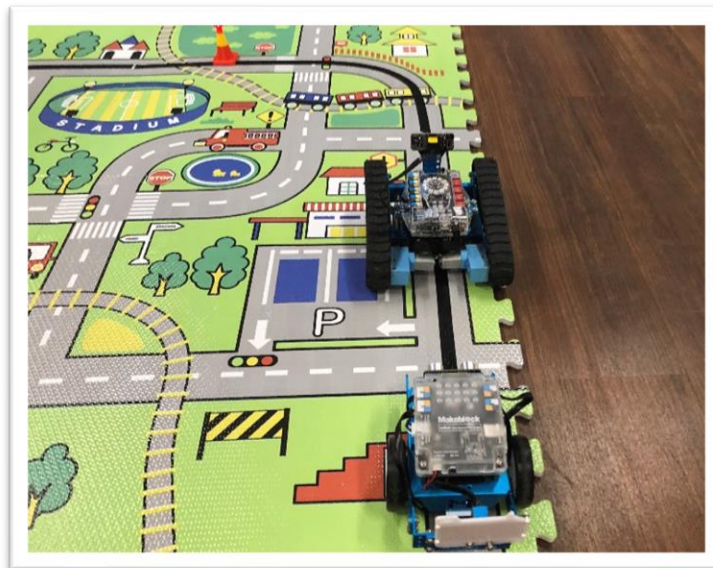
5.8.4. 7.^a e 8.^a aula - 30.03.2022

Na sétima e oitava aula efetuei inicialmente o registo de presenças e a revisão dos conteúdos lecionados na aula anterior, em concreto a paleta de cores RGB para obter cores compostas com os *LED's* do *robot*, que tinha suscitado algumas dúvidas nos alunos.

Na planificação inicial estava previsto ser apresentado o sensor de cor e o sensor de luz, mas infelizmente não foi possível de obter o sensor de cor, (nem por aquisição, nem por empréstimo), assim mais uma vez foi necessário repensar a planificação inicial e com conhecimento do Professor Cooperante alterei os conteúdos a lecionar nestas aulas.

Após a visualização da apresentação eletrónica (*vide* Anexo N), introduzi os novos conceitos e a respetiva associação das funcionalidades tecnológicas (sensores e atuadores) do robot *mBot*.

Figura 5.7 - Utilização de diferentes Robots no cenário



Os alunos idealizaram a ação do robot, com a utilização do sensor de luz e com o módulo *LED RGB*, retificando e melhorando a programação deste.

Por último submeteram as soluções do problema na plataforma da disciplina. Para utilização do sensor de luz foi necessário proceder à instalação de uma extensão (*Maker's Platform*) que contém os blocos de código para a programação deste sensor.

A aula terminou com uma breve síntese dos conteúdos lecionados.

5.8.5. 9.^a e 10.^o aula - 04.05.2022

Na nona aula, após o registo de presenças, foi efetuada a habitual revisão dos conteúdos lecionados na aula anterior.

Foi realizada a introdução dos novos conceitos relativos aos sensores a trabalhar na aula e também a associação das funcionalidades tecnológicas (sensores e atuadores) do robot *mBot* à resolução de problemas específicos, foi ainda utilizada uma apresentação eletrónica (*vide Anexo O*).

Figura 5.8 - Simulação de obstáculos no cenário



Para tal, os alunos tiveram de idealizar a ação do robot, com a utilização do sensor movimento PIR (*Passive Infrared Sensor*), retificar e melhorar a programação deste, por último submeter a solução na plataforma da disciplina.

Para utilização do sensor de luz foi necessário proceder à instalação da extensão *Maker's Platform*, (referida anteriormente) que contém os blocos de código para a programação deste sensor.

A aula terminou com uma breve síntese dos conteúdos lecionados.

Na décima aula, foram apresentadas, discutidas e avaliadas as soluções finais dos problemas propostas por cada grupo. Assim os alunos tiveram a oportunidade de ver as soluções apresentadas pelos diferentes grupos e verificar a programação efetuada por estes.

Foi ainda feita uma breve análise pelo Professor Estagiário do trabalho realizado durante estas 5 aulas da intervenção no âmbito da Prática de Ensino Supervisionada, realçando o empenho, e dedicação demonstrada pelos alunos nestes momentos de aprendizagem.

Finda a apresentação das tarefas os alunos efetuaram a sua autoavaliação, foi-lhes ainda pedido que procedessem ao preenchimento do inquérito avaliativo das atividades letivas que seria disponibilizado na plataforma da disciplina.

6. Dimensão Investigativa

Neste capítulo descrevo todo o processo de investigação realizado no decurso da Prática de Ensino Supervisionada, desde a metodologia de investigação, aos instrumentos e procedimentos utilizados para recolha e análise de dados.

Para tal, foram previamente elaboradas as respetivas autorizações, em formato de consentimento informado, para a realização da investigação, uma destinada ao Diretor do Agrupamento de Escolas de Portela e Moscavide (*vide* Anexo P) e a outra destinada aos Encarregados de Educação dos alunos (*vide* Anexo Q). Nestas os indicados deveriam manifestar o seu consentimento para realização/participação no estudo, sendo igualmente informados acerca do tipo de estudo a realizar, os seus propósitos, os procedimentos, ao mesmo tempo que lhes era garantido o anonimato e a confidencialidade dos dados pessoais.

Almeida & Freire, (2008), indicam que uma investigação é dirigida à testagem de hipóteses e que se pretende, ao definir um plano de investigação, criar as condições para que os dados obtidos sejam significativos para o problema em questão. Tal significância passa indubitavelmente pela qualidade das amostras tomadas, ou seja, junto de quem foi realizada a investigação.

A perspetiva quantitativa desenvolve a chamada metodologia hipotético-dedutiva, segundo a qual, a explicação causal se gere através de uma lógica dedutiva, ou seja, a pesquisa está referenciada a uma teoria que fundamenta e justifica as tentativas de explicação para as hipóteses de investigação.

Este estudo é de pequena dimensão devido ao número de intervenientes na prática de ensino supervisionada $n < 30$ e a amostra não foi constituída através de um processo de aleatorização.

Na investigação científica, por norma, recorre-se a amostras extraídas de populações às quais se querem depois generalizar os resultados. As amostras a tomar devem, então, possuir certas características como forma de se constituírem em boas representações das populações de onde foram retiradas (Almeida & Freire, 2008).

Assume-se como variável dependente a “aprendizagem da programação” e como a variável independente o “género”. A primeira foi operacionalizada através da aplicação de um Ficha de avaliação diagnóstica inicial (pré-teste) e uma Ficha de avaliação final (pós-teste); a segunda através da divisão dos alunos em dois grupos: rapazes e raparigas.

Quanto à análise foi efetuado um tratamento estatístico assente na comparação dos resultados das raparigas *versus* rapazes refletidos na avaliação sumativa dos problemas resolvidos em sala de aula.

Apresenta-se assim de seguida as hipóteses da presente investigação, sendo que se apresenta a hipótese zero e hipótese alternativa.

Hipótese de estudo 1:

- H0: Não existe diferenças significativas na pontuação apresentada pelos alunos na ficha aplicada no pré-teste e a apresentada no pós-teste;
- HAlt.: Existem diferenças significativas entre as pontuações que o grupo de alunos obteve na ficha aplicada no pré-teste e a apresentada no pós-teste.

Hipótese de estudo 2:

- H0: Não existe diferenças significativas na pontuação apresentada pelos grupos de rapazes e o grupo de raparigas nas pontuações obtidas nas fichas aplicadas;
- HAlt.: Existem diferenças significativas entre a pontuação apresentada pelos grupos de rapazes e o grupo de raparigas nas pontuações obtidas nas fichas aplicadas.

6.1. Instrumentos e procedimentos de Recolha de Dados

A recolha de dado foi efetuada através de fichas de avaliação desenvolvidas em paridade e aplicadas em dois momentos, uma Ficha de avaliação inicial (pré-teste) e uma Ficha de avaliação final (pós-teste). Ambos os testes foram aplicados em sala de aula, no início e no fim da intervenção, recorrendo à plataforma de avaliação SIMA (Sistema Interativo de Monitorização das Aprendizagens) disponibilizada pelo CCEMS (Centro de Competências Entre Mar e Serra) que permitiu aos alunos preencher online num qualquer dispositivo eletrónico (smartphone, tablet ou computador).

Tanto a Ficha de avaliação inicial (*vide* Anexo T) como a Ficha de avaliação final (*vide* Anexo U) eram compostas por 20 questões (num total de 100 pontos) divididas por domínios, sendo: 5 questões para o domínio da algoritmia, que valiam 25% da cotação total (25 pontos); 5 questões para o domínio da robótica que valiam 25% da cotação total (25 pontos) e 10 questões para o domínio da programação que

valiam 50% da cotação total (50 pontos). Na ficha de avaliação inicial as questões eram mais genéricas e menos complexas, para perceber os conhecimentos que cada aluno possuía sobre cada domínio. Na ficha de avaliação final as questões elaboradas incidiram mais sobre os conteúdos lecionados durante a intervenção para assim aferir a aquisição de conhecimentos dos alunos.

Estes dados recolhidos de forma organizada com resumos estatísticos permitiram ter uma visão geral do que foi melhor conseguido e do que não foi tão bem conseguido, assim com base nessas evidências procuro responder á questão de investigação colocada.

6.2. Resultados

Relativamente aos resultados devo salientar que idealmente para fazer análise de dados de natureza paramétrica, devemos de considerar idealmente três requisitos:

1. Um número considerável de participantes ($n > 30$ nos diferentes grupos);
2. Homogeneidade de variâncias;
3. Normalidade da distribuição.

O primeiro requisito não se cumpre dado que temos um $n_{total} = 25$, e também não se verificou o terceiro requisito da normalidade da distribuição, o que é compreensível dado que quanto menor é o número de observações, menor é a proximidade entre a sua distribuição e a curva de Gauss. Deste modo, optei pela realização de testes não paramétricos (*vide Anexos R e S*) para realizar análise comparativa de médias entre o grupo de rapazes e o grupo das raparigas, considerando os dois momentos de análise.

6.2.1. Análise comparativa pré-teste e pós-teste

Sabendo que os teste a realizar são testes não paramétricos, procedi a análise comparativa, comparando o desempenho no início e do fim do estudo.

Tabela 6.1 - Resultado da Estatística Descritiva do Teste de Wilcoxon

		N	Média	Erro Desvio
Pontos	pré-teste	25	64,800	12,0312
Pontos	pós-teste	25	69,600	10,5987
Algoritmia	pré-teste	25	12,60	6,474
Algoritmia	pós-teste	25	16,80	4,975
Robótica	pré-teste	25	16,00	6,455
Robótica	pós-teste	25	16,60	4,726
Programação	pré-teste	25	36,20	5,260
Programação	pós-teste	25	36,00	6,455
N válido (de lista)		25		

Verifica-se que existe melhoria nos resultados dos alunos em todas as dimensões estudadas do momento inicial do pré-teste para o pós-teste com exceção do domínio da programação, na medida em que no pós-teste os resultados são sempre superiores.

Para identificar a significância estatística dessas diferenças realizei então o teste Wilcoxon.

Tabela 6.2 - Resultado do resumo de Teste de Hipótese (pontos)

Resumo de Teste de Hipótese				
	Hipótese nula	Teste	Sig.	Decisão
1	A mediana de diferenças entre Pontos e Pontos é igual a 0.	Amostras Relacionadas de Teste dos Postos Sinalizados de Wilcoxon	,001	Rejeitar a hipótese nula.

São exibidas significâncias assintóticas. O nível de significância é ,050.

Pela análise da tabela 6.2, verifica-se que o resultado (decisão) é a de “Rejeitar a hipótese nula”, ou seja, há diferenças entre os grupos, os alunos apresentaram melhor desempenho no fim do que apresentaram no início, quanto aos resultados dos pontos do teste (*vide* Tabela 6.1).

Tabela 6.3 – Resultado do resumo de Teste de Hipótese (algoritmia)

Resumo de Teste de Hipótese				
	Hipótese nula	Teste	Sig.	Decisão
1	A mediana de diferenças entre Algoritmia e Algoritmia é igual a 0.	Amostras Relacionadas de Teste dos Postos Sinalizados de Wilcoxon	<,001	Rejeitar a hipótese nula.

São exibidas significâncias assintóticas. O nível de significância é ,050.

Pela análise da tabela 6.3, verifica-se que relativamente à algoritmia, o resultado (decisão) é a de “Rejeitar a hipótese nula”, ou seja, há diferenças entre os grupos, eles apresentaram melhor desempenho no fim do que apresentaram no início, quanto aos resultados da algoritmia do teste (*vide* Tabela 6.1).

Tabela 6.4 - Resultado do resumo de Teste de Hipótese (robótica)

Resumo de Teste de Hipótese				
	Hipótese nula	Teste	Sig.	Decisão
1	A mediana de diferenças entre Robótica e Robótica é igual a 0.	Amostras Relacionadas de Teste dos Postos Sinalizados de Wilcoxon	,509	Reter a hipótese nula.

São exibidas significâncias assintóticas. O nível de significância é ,050.

Pela análise da tabela 6.4, verifica-se que o resultado (decisão) é a de “Reter a hipótese nula”, ou seja, as diferenças entre os grupos, não foram significativas, quanto aos resultados obtidos para a seção do teste relativa à robótica no teste de diagnóstico comparativamente ao teste final (*vide* Tabela 6.1).

Tabela 6.5 - Resultado do resumo de Teste de Hipótese (programação)

Resumo de Teste de Hipótese				
	Hipótese nula	Teste	Sig.	Decisão
1	A mediana de diferenças entre Programação e Programação é igual a 0.	Amostras Relacionadas de Teste dos Postos Sinalizados de Wilcoxon	,941	Reter a hipótese nula.

São exibidas significâncias assintóticas. O nível de significância é ,050.

De igual modo na tabela 6.5, verifica-se que o resultado (decisão) é a de “Reter a hipótese nula”, ou seja, as diferenças entre os grupos não mostraram ser significativas quanto aos resultados da programação obtidos no teste inicial vs. final (*vide* Tabela 6.1).

Desta forma, verifica-se que só se detetaram diferenças estatisticamente significativas entre o início do estudo e o fim do estudo, naquilo que foram os resultados globais, mas igualmente e de forma específica no que diz respeito à robótica e à programação as diferenças não foram estatisticamente significativas, nem na primeira onde elas melhoraram, nem na segunda onde até se registou algum retrocesso.

6.2.2. Análise comparativa por género

Nesta análise em específico como se constituíram grupos independentes (quem está no grupo dos rapazes não está no grupo das raparigas), realizei os testes não-paramétricos de comparação de média para amostras independentes.

Para o caso foi efetuada primeiramente uma análise da estatística descritiva da comparação entre os resultados obtidos (médias) pelos rapazes e pelas raparigas no teste final.

Tabela 6.6 - Resultado da Estatística Descritiva das médias (Teste-T amostras independentes)

Estatísticas de grupo					
	Sexo 1 masculino 2 feminino	N	Média	Erro Desvio	Erro padrão da média
Pontos	masculino	10	63,182	10,3133	3,1096
pré-teste	feminino	8	63,125	11,3192	4,0020
Pontos	masculino	10	67,273	8,4746	2,5552
pós-teste	feminino	8	70,625	12,3744	4,3750
Algoritmia	masculino	10	13,64	6,360	1,918
pré-teste	feminino	8	10,00	7,071	2,500
Algoritmia	masculino	10	15,91	5,394	1,626
pós-teste	feminino	8	18,75	4,432	1,567
Robótica	masculino	10	15,00	5,477	1,651
pré-teste	feminino	8	16,25	7,906	2,795
Robótica	masculino	10	15,00	5,477	1,651
pós-teste	feminino	8	17,50	4,629	1,637
Programação	masculino	10	34,55	5,222	1,575
pré-teste	feminino	8	36,88	3,720	1,315
Programação	masculino	10	36,36	4,523	1,364
pós-teste	feminino	8	33,75	8,345	2,950

Como o número de rapazes na turma é substancialmente superior ao número de raparigas foi efetuada uma seleção aleatória de rapazes para se constituir um grupo numericamente equivalente ao das raparigas e assim se efetuar a comparação entre grupos, seguindo a regra **maior grupo = menor grupo * 1.4**, de acordo com Pestana & Gageiro (2003).

Tabela 6.7 - Resultado do Teste Não Paramétrico (Mann-Whitney)

Resumo de Teste de Hipótese				
	Hipótese nula	Teste	Sig.	Decisão
1 pré-teste	A distribuição de Pontos é igual nas categorias de Sexo 1 masculino 2 feminino.	Amostras Independentes de Teste U de Mann-Whitney	,968 ^a	Reter a hipótese nula.
2 pós-teste	A distribuição de Pontos é igual nas categorias de Sexo 1 masculino 2 feminino.	Amostras Independentes de Teste U de Mann-Whitney	,492 ^a	Reter a hipótese nula.
3 pré-teste	A distribuição de Algoritmia é igual nas categorias de Sexo 1 masculino 2 feminino.	Amostras Independentes de Teste U de Mann-Whitney	,272 ^a	Reter a hipótese nula.
4 pós-teste	A distribuição de Algoritmia é igual nas categorias de Sexo 1 masculino 2 feminino.	Amostras Independentes de Teste U de Mann-Whitney	,351 ^a	Reter a hipótese nula.
5 pré-teste	A distribuição de Robótica é igual nas categorias de Sexo 1 masculino 2 feminino.	Amostras Independentes de Teste U de Mann-Whitney	,545 ^a	Reter a hipótese nula.
6 pós-teste	A distribuição de Robótica é igual nas categorias de Sexo 1 masculino 2 feminino.	Amostras Independentes de Teste U de Mann-Whitney	,310 ^a	Reter a hipótese nula.
7 pré-teste	A distribuição de Programação é igual nas categorias de Sexo 1 masculino 2 feminino.	Amostras Independentes de Teste U de Mann-Whitney	,492 ^a	Reter a hipótese nula.
8 pós-teste	A distribuição de Programação é igual nas categorias de Sexo 1 masculino 2 feminino.	Amostras Independentes de Teste U de Mann-Whitney	,600 ^a	Reter a hipótese nula.

São exibidas significâncias assintóticas. O nível de significância é ,050.

a. A exata significância é exibida para este teste.

Verificou-se, que para a totalidade das dimensões em estudo, tanto para a totalidade do teste como para as seções (algoritmia, robótica e programação) o resultado é de “Reter a hipótese nula”, ou seja, não existem diferenças estatisticamente significativas entre os dois grupos (rapazes e raparigas).

Verifica-se que ambos os grupos do sexo masculino e feminino têm resultados idênticos, posso verificar que em algumas variáveis eles revelam melhor desempenho, noutros elas são melhores. Assim esta diferença, trazida recorrentemente como existente na literatura, não se verificou no presente estudo.

6.3. Componente Investigativa

A dimensão de investigação foi realizada durante a intervenção e visou a compreensão da problemática relativa ao impacto das dificuldades dos alunos na aprendizagem da programação.

Esta questão torna-se cada vez mais pertinente numa sociedade que promove a igualdade de género mas onde, no entanto, se verifica ainda que numa área fundamental como as *STEM* (*Science, Technology, Engineering e Mathematics*) o número de homens com bacharelato e licenciaturas na área continua a superar largamente o número de mulheres, situação esta que depois se reflete nos postos de trabalho (Hill et al., 2010).

Tendo em conta o tamanho do estudo, que devido aos seus intervenientes (25 alunos) é considerado um estudo de pequena dimensão, com uma amostra não aleatória, faz com que os dados recolhidos devam ser interpretados com cautela, no entanto, apesar de todas estas condicionantes ainda é possível de responder às questões de investigação colocadas. Assim, e mediante os resultados do estudo, verifica-se que:

- Quanto à hipótese de estudo 1, rejeita-se H_0 e aceita-se H_{Alt} , existem diferenças significativas entre as pontuações que o grupo de alunos obteve na ficha aplicada no pré-teste e a apresentada no pós-teste, verifica-se melhoria dos resultados obtidos;
- Quanto à hipótese de estudo 2, rejeita-se H_{Alt} e aceita-se H_0 , não existindo, pois, diferenças significativas na pontuação apresentada pelos grupos de rapazes e o grupo de raparigas nas fichas aplicadas.

Com base na análise efetuada, e recuperando-se a questão de investigação feita conclui-se não existirem diferenças significativas na aprendizagem da programação de robots entre os dois grupos do estudo, rapazes e raparigas.

6.4. Avaliação das Aula Lecionadas

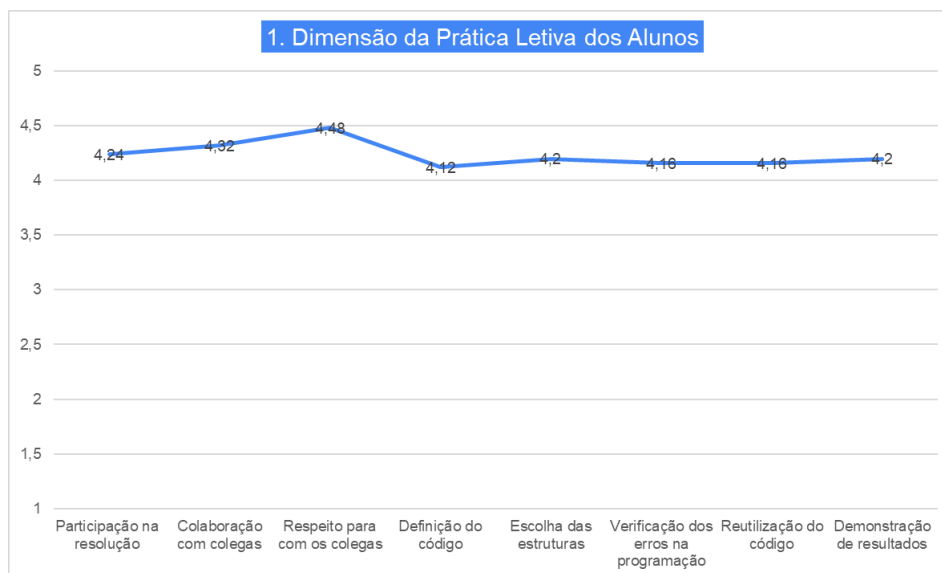
No final da intervenção, no último dia foi efetuado um questionário (*vide* Anexo Y) que visou perceber a compreensão das seguintes dimensões: 1. Dimensão da Prática Letiva dos Alunos; 2. Dimensão do Processo de Ensino e Aprendizagem; 3. Dimensão da Autoavaliação dos Alunos.

Assim no ponto 1. Dimensão da Prática Letiva dos Alunos, foram colocadas as seguintes perguntas:

- Participaste ativamente na resolução dos problemas propostos?
- Colaboraste com os teus colegas na resolução dos problemas?
- Respeitaste os direitos, as opiniões e capacidades dos teus colegas no decurso do trabalho colaborativo em sala de aula?
- Procuraste definir corretamente a estrutura dos programas elaborados?
- Escolheste criteriosamente as estruturas de controlo (seleção e/ou repetição) necessárias para resolver os problemas?
- Verificaste e corrigiste os erros de programação?
- Reutilizaste o código sempre que foi necessário e apropriado?

- Demonstraste a compreensão dos programas e dos resultados produzidos?

Figura 6.1 - Gráfico 1. Dimensão da Prática Letiva dos Alunos



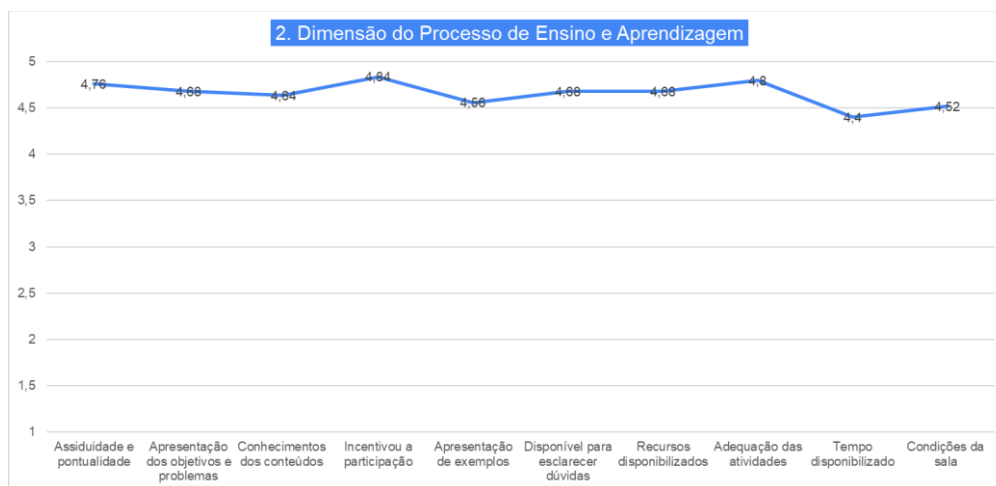
Os resultados apresentados no gráfico (*vide* Figura 6.1) demonstram que as respostas dos alunos demonstram uma avaliação genericamente positiva relativamente à sua prática letiva, sendo as respostas superiores a 4 valores.

No ponto 2. Dimensão do Processo de Ensino e Aprendizagem, foram colocadas as seguintes perguntas:

- O Professor foi assíduo e pontual?
- O Professor apresentou os objetivos de aprendizagem e os problemas de uma forma clara e objetiva?
- O Professor revelou conhecimento sobre os conteúdos abordados na intervenção?
- O Professor incentivou a participação dos alunos na resolução dos problemas?
- O Professor apresentou exemplos de projetos com diferentes graus de dificuldades?
- O Professor mostrou-se disponível para o esclarecimento de dúvidas?
- O Professor disponibilizou os recursos suficientes na plataforma "Google Classroom" (apresentações eletrónicas, guiões, vídeos, ...)?

- As atividades realizadas foram adequadas para a aprendizagem dos conteúdos?
- O tempo disponibilizado para cada problema foi suficiente?
- A sala reúne as condições para o desenvolvimento da aprendizagem da Robótica Educativa?

Figura 6.2 - Gráfico 2. Dimensão do Processo de Ensino e Aprendizagem



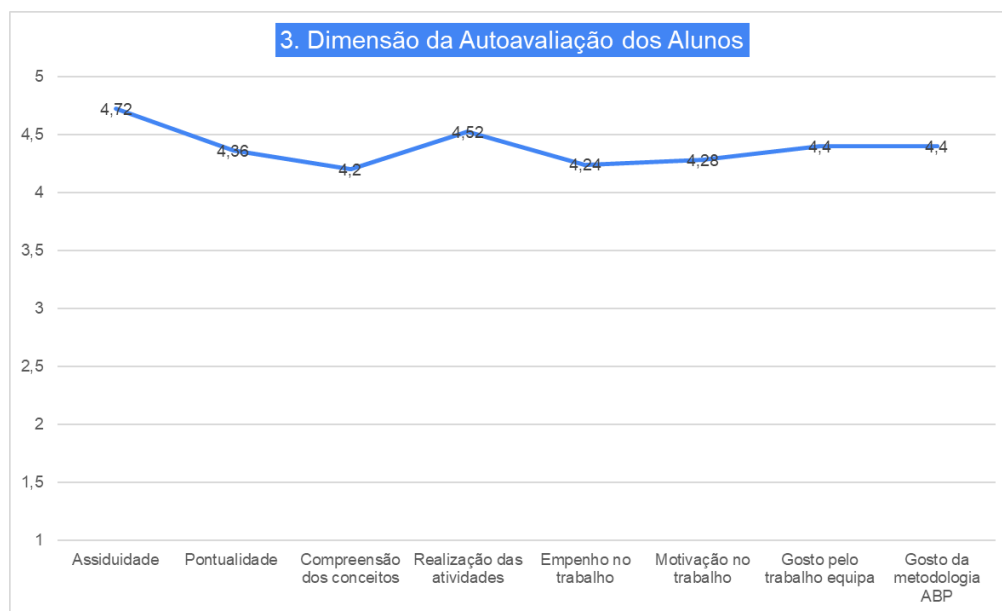
Os resultados apresentados no gráfico (*vide* Figura 6.2) demonstram que as respostas dos alunos foram genericamente positivas relativamente ao processo de ensino e aprendizagem, sendo a média das respostas superior a 4,5 valores.

No ponto 3. Dimensão da Autoavaliação dos Alunos, foram colocadas as seguintes perguntas:

- Foste assíduo?
- Foste pontual?
- Compreendeste os conceitos científicos apresentados?
- Realizaste todas as tarefas propostas nas atividades letivas?
- Estiveste empenhado na realização dos problemas propostos nas atividades letivas?
- Estiveste motivado na realização dos problemas propostos durante as atividades letivas?
- Gostaste de trabalhar em equipa (colaborativamente) na resolução dos problemas propostos?

- Gostaste de trabalhar com a metodologia de Aprendizagem Baseada em Problemas?

Figura 6.3 - Gráfico 3. Dimensão de Autoavaliação dos Alunos



Os resultados apresentados no gráfico (*vide* Figura 6.3) demonstram que as respostas dos alunos mais uma vez revelam uma avaliação genericamente positiva relativamente à sua autoavaliação, sendo as respostas todas superiores a 4,2 valores. Assim feita uma apreciação global dos resultados deste inquérito sobre a intervenção, é possível verificar que os resultados aqui representados têm uma média superior a 4 valores, na escala utilizada que variava entre 1 e 5 valores, não existindo assim nenhuma apreciação negativa das dimensões representadas.

7. Conclusão

Este relatório, foi elaborado no âmbito da unidade curricular de Iniciação à Prática Profissional IV, seguindo a planificação da Prática de Ensino Supervisionada que se realizou no âmbito da unidade curricular de Iniciação à Prática Profissional III. A mesma decorreu na Escola Secundária da Portela entre fevereiro e maio de 2022, com o Professor Cooperante Paulo Torcato na turma B do 9.º ano, na disciplina de Introdução à Robótica, módulo 3 - Introdução à Programação com o recurso à Robótica.

A intervenção foi planificada para ter a duração de 500 minutos. Inicialmente a intervenção estava prevista decorrer entre 23 de fevereiro e 23 de março, no entanto, devido a imprevistos (descritos anteriormente), acabou por prolongar a intervenção até ao dia 4 de maio de 2022.

Na planificação inicial das atividades letivas, foi necessário proceder alguns reajustes, devido à impossibilidade em adquirir alguns componentes eletrónicos (sensores) que estavam previstos ser utilizados e foi necessário substituí-los por outros e refazer os respetivos planos de aula e as apresentações eletrónicas (aulas 5/6 e 7/8).

Quanto aos resultados dos alunos, inicialmente verifiquei que existiu uma certa dificuldade em dominar os conceitos teóricos de algoritmia e programação, mas com todos os recursos disponibilizados na plataforma da disciplina (documentos de apoio) e a revisão sistemática dos conteúdos onde demonstraram dúvidas ao longo das atividades letivas, foi possível ultrapassar essas dificuldades. Para isso, o acompanhamento nos grupos de trabalho foi muito importante para ganhar a confiança dos alunos, ajudar quando necessário e incentivar a participação de cada um nas tarefas propostas. A partir da segunda aula, a entajuda existente entre os membros do grupo aumentou, sobretudo para com aqueles que mais dificuldade demonstravam. No fim penso que todo esse empenho por parte de todos os intervenientes (alunos, Professor Estagiário e Professor Cooperante) se refletiu nos resultados obtidos e no sucesso académico alcançado.

Quanto às hipóteses que apresento, existe bastante literatura científica e investigação realizada, sobre as diferenças de género no acesso à área da STEM, e estes apontam várias causas, Cvencek, Meltzoff, e Greenwald (2011, citado por Master et al., 2015) indicam que quando as raparigas chegam à adolescência são alvo de estereótipos negativos acerca das suas capacidades para a matemática e para a ciência.

Já Goode, Estrella, & Margolis, (2006) e Mercier et al., (2006 como citado em Master et al., 2015) referem que outro fator influenciador, é que as áreas STEM são dominadas pelos homens. Por sua vez, Cheryan et al., (2015) afirmam que a ausência de mulheres na área das STEM deve-se às restrições na sua escolha devido particularmente aos estereótipos sobre o tipo de pessoas que devem trabalhar nestas áreas, ao trabalho envolvido e aos valores dessas áreas.

Master et al. (2017) indicam que pesquisas recentes sugerem que uma experiência positiva precoce com a programação pode levar a uma maior motivação no campo da robótica e da programação para as raparigas, nomeadamente quando se compara estas com outras raparigas sem essas experiências. Professores, pais e decisores que criam experiências STEM positivas para as raparigas têm o potencial de colocar as mesmas em trajetórias académicas que podem levar a uma maior participação nas ciências da computação e engenharias. Por sua vez, Metz, (2007) e Steele, (1997 citado por Sullivan & Bers, 2016) sugerem que uma das maneiras de lidar com essa disparidade de género é atrair o interesse das meninas logo nos inícios da escolaridade (pré-escolar), antes que os estereótipos de géneros sejam enraizados.

No caso em particular do estudo realizado nesta intervenção verifiquei que uma grande parte das raparigas da turma já tinham sido alunas do Professor Cooperante no 1.º e 2.º ciclo, tendo frequentado as aulas de robótica como oferta complementar regularmente. Entende-se que este facto tenha sido um fator importante para que neste estudo não existissem grandes diferenças entre os rapazes e as raparigas nos resultados dos testes efetuados.

7.1. Balanço Reflexivo

A frequência deste Mestrado, além ter sido uma experiência árdua em virtude de ter frequentado este ciclo de estudos como trabalhador-estudante e este decorrer em regime diurno e presencial, também teve momentos de aprendizagem relevantes e de enorme importância pedagógica. Posso destacar como exemplo, o desenvolvimento e consolidação das competências previstas na legislação deste curso (Regulamento n.º 553/2017), como: i) aplicar os conhecimentos e a capacidade de compreensão e resolução de problemas em situações novas e multidisciplinares dentro da minha área de estudo; ii) a capacidade de integrar conhecimentos e lidar com situações complexas; iii) desenvolver soluções e incluir reflexões sobre as implicações e responsabilidade

éticas; iv) a capacidade de comunicar as minhas conclusões, conhecimentos e raciocínios; v) possuir competências que permitam uma aprendizagem ao longo da vida de um modo auto-orientado ao autónomo. Estas competências dificilmente poderiam ser trabalhadas fora deste contexto na medida em que requerem efetivamente investimento intelectual e preparação de natureza académica.

Por último, tenho de salientar a importância das várias disciplinas deste curso tanto da Faculdade de Ciências da Universidade de Lisboa (FCUL) como as do Instituto de Educação (IE), e devo destacar pelos menos as mais impactantes que foram as disciplinas de Didática da Informática e de Introdução à Prática Pedagógica pela intensidade destas, pelos conteúdos estudados, pelos trabalhos desenvolvidos e pela importância reconhecida às mesmas neste meu percurso formativo. Não menos importante, a importância dos docentes em especial os docentes orientadores da FCUL na pessoa do Professor Doutor Luís Moniz e do IE na pessoa da Professora Doutora Neuza Pedro, que foram dedicados, persistentes, incansáveis e uma referência como docentes que dificilmente esquecerei. Não menos importante o Professor Paulo Torcato, que demonstrou ser um verdadeiro timoneiro nesta viagem de aprendizagem e de descoberta do conhecimento. A todos o meu reconhecimento e agradecimento pela vossa disponibilidade, foram sem dúvida fundamentais para o sucesso neste percurso e para a minha evolução como aluno e futuramente como docente.

Na planificação desta intervenção colocaram-se vários desafios como o facto de usar uma metodologia da Aprendizagem Baseadas em Problemas para o ensino da Robótica Educativa e conseqüentemente para a aprendizagem da programação por parte dos alunos, compreendi a importância dos momentos de aprendizagem em sala de aula, sejam alcançados através de métodos de aprendizagem ativa, momentos que envolvam os alunos, que promovam a discussão, o debate, que sejam momentos de partilha e de saberes entre todos os intervenientes.

Numa área tão abrangente como a robótica educativa penso que 500 minutos de intervenção são insuficientes para a quantidade de conteúdos que poderiam ser abordados, dado que esta é uma área sempre em atualização e desenvolvimento constante. Ainda assim procurei ser bastante assertivo e selecionar um conjunto de conteúdos que seriam, na minha perceção os mais indicados para promover uma aprendizagem concreta dos conteúdos propostos.

A dinâmica de trabalho desenvolvida foi muito intensa, mas foi interessante ver o envolvimento, a dedicação dos alunos na programação dos robots e na resolução

dos problemas propostos, claro que começasse hoje faria alguns ajustes na planificação, mas isso faz parte do processo natural de evolução, perceber o que nem sempre corre tão bem e procurar melhorar a prática letiva por forma que seja mais interessante para os alunos.

Um dos reparos que tenho de fazer foi sobre os equipamentos disponíveis em sala de aula, apesar de termos alguns equipamentos muito bons, como o quadro interativo, impressoras 3D, tablets, dispositivos programáveis, tenho que mencionar que os computadores eram já antigos e acusavam o desgaste natural de muito uso. O facto de não serem máquinas recentes, acusavam o peso da idade e do uso, o que provocava alguma instabilidade e entraves ao desempenho. De igual modo, também os robots acusavam o desgaste de vários anos de utilização, penso que estes teriam cerca de 6 anos, por vezes apresentavam falhas (um motor com menos força, um sensor que não respondia, baterias que perdiam a carga rapidamente, etc.). Assim, foi necessário estar sempre vigilante e atento para intervir nos mesmos sempre que era necessário e não levar os alunos a pensar que as falhas estavam na programação efetuada. No entanto, afirmo que repetiria tudo outra vez, tal foi o prazer que me deu fazer esta intervenção, junto do Professor Cooperante que foi incansável e realmente justo perante o termo que o descreve: cooperante.

Termino afirmando que foi uma experiência única e um momento extraordinário de aprendizagem, no qual tive a felicidade de participar. Esta é a merecida conclusão para este trabalho e para os dois anos de formação que lhe tiveram por detrás.

8. Referências

- AEPM. (2021). *Documento Orientador da organização do ano letivo 2021/2022* (p. 34). <http://agepm.pt/cms/agrupamento/documento-orientador>
- AEPM a. (2020). *Projeto Educativo do Agrupamento de Escolas de Portela e Moscavide* (p. 48). <http://agepm.pt/cms/agrupamento/projeto-educativo>
- AEPM b. (2020). *Regulamento Interno do Agrupamento de Escolas de Portela e Moscavide - 2020/2024*. <http://agepm.pt/cms/agrupamento/regulamentos>
- Almeida, L., & Freire, T. (2008). *Metodologia da Investigação em Psicologia e Educação* (5ª). Edições, Psiquilíbrios.
- Baudson, A., & Araújo, F. (2013). *Algoritmos e Programação*. Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais.
- Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. Em *Computers and Education* (Vol. 72, pp. 145–157). <https://doi.org/10.1016/j.compedu.2013.10.020>
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Annual American Educational Research Association meeting, Vancouver, BC, Canada*, 1–25.
- Chalmers, C. (2018). Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction*, 17, 93–100. <https://doi.org/10.1016/j.ijcci.2018.06.005>
- Cheryan, S., & Master, A. (2015). Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering by diversifying stereotypes. *Frontiers in Psychology*, 6(FEB), 1–8. <https://doi.org/10.3389/fpsyg.2015.00049>
- Dorotea, N. (2013). Avaliação Online Das Aprendizagens Com Propósitos Formativos: Nota Positiva? Em *Mestrado em Tecnologias e Metodologias em E-Learning: Vol. I*. http://dx.doi.org/10.1007/978-3-540-88682-2_51
- Dorotea, N., & Pedro, N. (2015). *Provas digitais online na avaliação formativa . Exploração das práticas e conceções dos professores*. In *Atas da*(March), 484–497.

- Farrer, H. (1999). *Algoritmos Estruturados: Programação Estruturada de Computadores* (LTC (ed.)).
- Fernandes, E. (2013). Aprender matemática e informática com robots. Em *Book*. Universidade da Madeira. http://www.cee.uma.pt/droide2/ebook/ebook_vf.pdf
- Ferreira, F., Veruggio, G., Micheli, E., & Operto, F. (2010). *The Proliferation of Educational Robotics*.
- Gomes, A., Henriques, J., & Mendes, A. (2008). Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. *Educação, Formação & Tecnologias - ISSN 1646-933X*, 1(1), [93-103].
- Hehl, M. (1986). *Linguagem de Programação Estruturada FORTRAN 77*. McGraw-Hill.
- Hill, C., Corbett, C., & St Rose, A. (2010). Why So Few ? Why So Few ? Em AAUW - American Association of University (Ed.), *American Association of University Women* (Vol. 5, Número 3). <http://eric.ed.gov/ERICWebPortal/recordDetail?accno=ED509653>
- Jenkins, T. (2002). On the Difficulty of Learning To Program. *3rd Annual LTSN-ICS Conference, Loughborough University*, 53–58. <http://www.comp.leeds.ac.uk/tony/>
- Junior, J., & Rapkiewicz, C. (2005). Um Ambiente Virtual para apoio a uma Metodologia para Ensino de Algoritmos e Programação. *Renote*, 3(2), 1–7. <https://doi.org/10.22456/1679-1916.14021>
- Martins, J. P. (2013). *Programação em Python: Introdução à Programação Utilizando Múltiplos Paradigmas*. Instituto Superior Técnico.
- Master, A., Cheryan, S., Moscatelli, A., & Meltzoff, A. (2017). Programming experience promotes higher STEM motivation among first-grade girls. *Journal of Experimental Child Psychology*, 160, 92–106. <https://doi.org/10.1016/j.jecp.2017.03.013>
- Master, Al., Cheryan, S., & Meltzoff, A. (2015). Computing Whether She Belongs: Stereotypes Undermine Girls' Interest and Sense of Belonging in Computer Science. *Journal of Educational Psychology*, 57. <https://doi.org/https://dx.doi.org/10.1037/edu0000061>
- Matos, J., Pedro, A., & Piedade, J. (2019). Learning scenarios in computer science

- initial teacher education. *Revista Lusofona de Educacao*, 45(45), 223–238.
<https://doi.org/10.24140/issn.1645-7250.rle45.15>
- Medina, M., & Fertig, C. (2005). *Algoritmos e Programação: Teoria e Prática* (N. Editora (ed.)).
- Oliveira, B. (2008). *Plataforma Robótica Multi-Funcional*. Universidade de Aveiro.
- Pedro, A., Matos, J., Piedade, J., & Dorotea, N. (2017). *Probótica - Programação e Robótica no Ensino Básico - Linhas Orientadoras*. 37.
- Pedro, A., Piedade, J., & Matos, J. (2019). Learning scenarios in computer science initial teacher education. *Revista Lusofona de Educacao*, 45(45), 223–238.
<https://doi.org/10.24140/issn.1645-7250.rle45.15>
- Pestana, M., & Gageiro, J. (2003). *Análise de dados para ciencias sociais: A complemen- taridade de SPSS [Data Analysis for Social Sciences: The Complementarity of SPSS]* (3rd ed.). Edições Silabo.
- Piedade, J. (2021). Pensamento Computacional, Programação e Robótica na Educação Básica: Experiências Didáticas na Formação de Professores. *LEITURA E ESCRITA NO MUNDO DIGITAL: desafios e oportunidades para alunos e professores, February*, 137–153.
- Piedade, J., Dorotea, N., Ferrentini, F., & Pedro, A. (2019). A cross-analysis of block-based and visual programming apps with computer science student-teachers. *Education Sciences*, 9(3). <https://doi.org/10.3390/educsci9030181>
- Reis, N. (2019). *Carta de Missão do Diretor do Agrupamento de Escolas de Portela e Moscavide* (p. 6). http://agepm.pt/cms/images/19-20/Agrupamento/Carta_de_Missao_do_Diretor_2019-2023.pdf
- Reis, P. (2011). *Observação de aulas e avaliação do desempenho docente*.
- Saeli, M., Perrenet, J., Jochems, W. M. G., & Zwaneveld, B. (2011). Teaching programming in secondary school: A pedagogical content knowledge perspective. *Informatics in Education*, 10(1), 73–88.
<https://doi.org/10.15388/infedu.2011.06>
- Santos, J. (2006). *Algoritmia e Programação*. Instituto Superior de Engenharia do Porto.
- Sullivan, A., & Bers, M. U. (2016). Girls, boys, and bots: Gender differences in young children's performance on robotics and programming tasks. *Journal of*

Information Technology Education: Innovations in Practice, 15(1), 145–165.

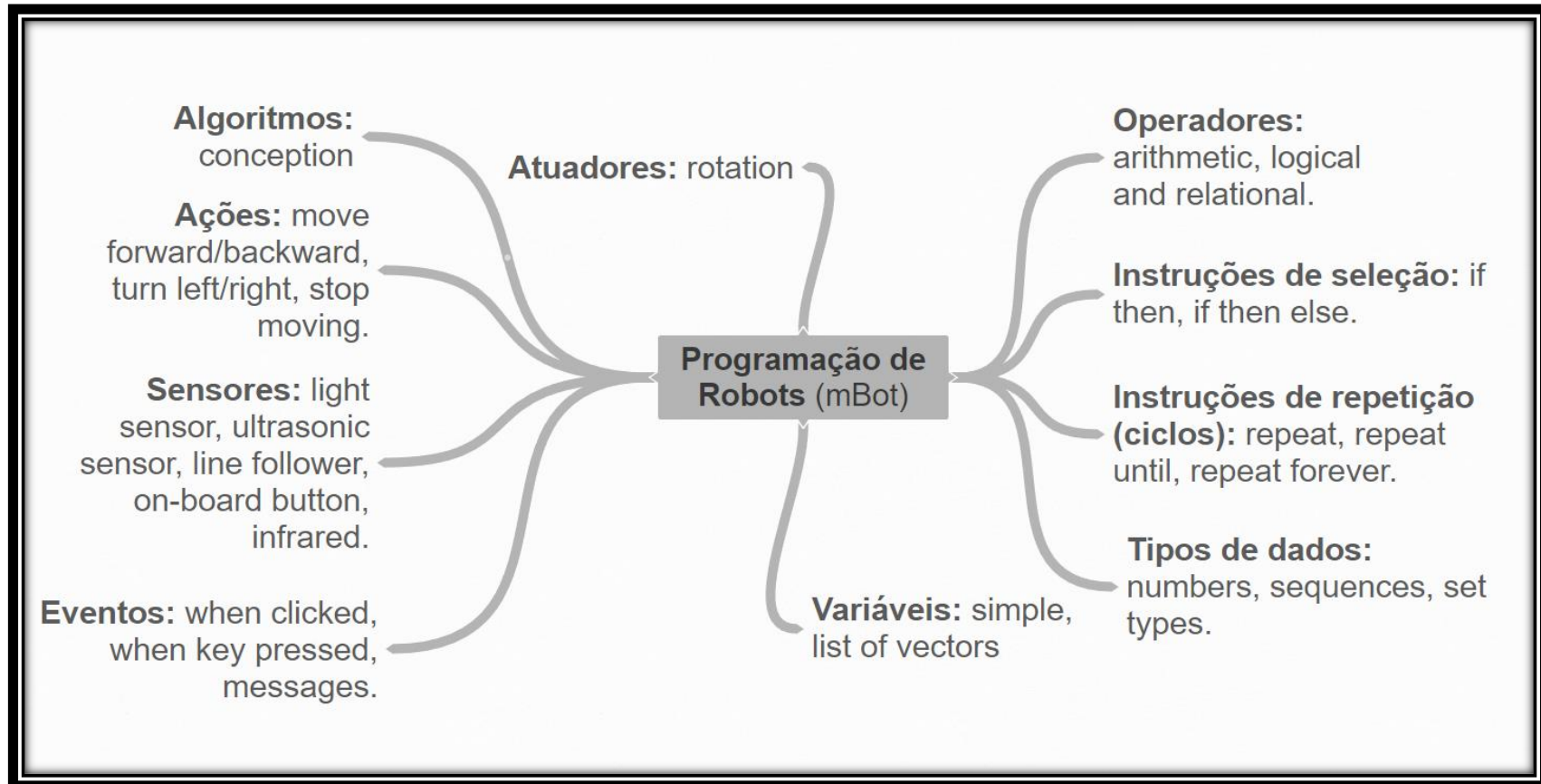
<https://doi.org/10.28945/3547>

Wing, J. (2006). Computational Thinking. Em *Communications of the ACM* (Vol. 49, Número 3, pp. 33–35). ACM.

Wing, J. (2011). Research notebook: Computational thinking-What and why? *The Link Magazine*, 1–8. <http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>

9. Anexos

Anexo A. Mapa de Conceitos Científicos da Programação de Robots mBot



Anexo B. Grelha de observação – Estratégias de Ensino - Professor

Nome do professor: Paulo Torcato	Ano: 9.º	Turma: B	
Disciplina: Introdução à Robótica	N.º alunos: 27	Hora: 09:55	
Observador: João Freitas	Sala: C03	Data:	
Comportamentos com impactos educativos positivos	Nada evidente	Algo evidente	Bem evidente
1. Expressa-se muito bem, tanto oralmente como por escrito.			
2. Fornece instruções de forma clara e concisa.			
3. Utiliza eficazmente as experiências, as ideias e os conhecimentos prévios dos alunos.			
4. Estimula e encoraja a participação dos alunos.			
5. Explica os conteúdos difíceis de mais de uma maneira.			
6. Utiliza diversas actividades na aula.			
7. Apresenta exemplos e demonstrações de determinados conteúdos.			
8. Utiliza meios audiovisuais diversos.			
9. Capta a atenção dos alunos.			
10. Reage e adapta-se às alterações de atenção dos alunos.			
11. Evidencia um entusiasmo sincero pelo tema da aula.			
12. Adequa as estratégias de ensino aos conteúdos.			
13. Adequa as estratégias de ensino à idade e as necessidades dos alunos.			
14. Demonstra excelente conhecimento dos conteúdos.			
15. Proporciona oportunidades aos alunos para que apliquem os conhecimentos.			
16. Estabelece relações entre os novos tópicos da aula e os tópicos já conhecidos.			
17. Utiliza textos na aula.			
18. Estabelece relações entre os tópicos da aula e os tópicos das aulas anteriores e posteriores.			
19. Atribui aos alunos tempo adequado para responderem às perguntas.			
20. Termina com as distrações dos alunos de forma construtiva.			
Comentários gerais:			

Anexo C. Grelha de observação – Dinamização da Aula - Alunos

Nome do aluno:	Ano: 9.º	Turma: B	
Disciplina: Introdução à Robótica	N:	Sala: C03	
Observador: João Freitas	Hora: 09:55	Data:	
Indicadores de evidências	Nada evidente	Algo evidente	Bem evidente
1. Expressa-se muito bem, tanto oralmente como por escrito acerca dos conceitos.			
2. Demonstra um comportamento correto e não interrompe as atividades de forma inapropriada.			
3. Demonstra capacidade de trabalho e adaptação às tarefas.			
4. Apresenta-se como aprendiz, indicando o que não sabe e mostrando satisfação quando aprende algo de novo.			
5. Envolve-se na tentativa de obter resposta para as perguntas colocadas no âmbito de uma atividade.			
6. Demonstra curiosidade pelas atividades propostas.			
7. Manipula corretamente materiais e equipamentos.			
8. Consegue desenvolver as tarefas dentro do tempo previsto.			
9. Trabalha em pares ou em pequenos grupos.			
10. Reage e adapta-se às alterações de atenção dos colegas.			
11. Evidencia um entusiasmo sincero pelo tema da aula.			
12. Demonstra raciocínio e aprendizagem acerca dos conceitos abordados nas atividades.			
13. Analisa e avalia a informação recolhida.			
14. Usa as ferramentas eletrónicas da escola (<i>classroom</i>), para comunicar informação relacionada com o trabalho.			
15. Demonstra as conceções prévias e trabalha-as individualmente ou com os seus pares.			
16. As respostas evidenciam raciocínio sobre os conceitos abordados.			
17. Procura oportunidades para aplicar as novas aprendizagens ao mundo real.			
18. Efetua pesquisas colaborativas no âmbito das atividades escolares.			
19. Evidencia capacidade de análise crítica da informação obtida.			
20. O aluno possui capacidade para a seleção das ferramentas tecnológicas mais adequadas a cada tarefa.			
Comentários gerais:			

Cenário de Aprendizagem “Smart City”



Disciplina: Introdução à Robótica
Módulo/ Unidade didática: Introdução à Programação
Turma: 9.º B
Autor: João Freitas

Breve descrição

Em que disciplina e respetiva temática se inscreve este cenário? De que modo este contribui para o desenvolvimento das competências preconizadas na disciplina?

UC: Introdução à Robótica, na unidade temática, Introdução à Programação.

Este cenário desenvolve-se na disciplina de oferta de escola de Introdução à Robótica, que na sua génese pretende trabalhar a aprendizagem das linguagens de programação através da robótica, proporcionando assim um ensino de índole mais prático e colaborativo. Os conteúdos da unidade temática a trabalhar relacionam-se com a introdução à programação.

O cenário aqui apresentado inspira-se nas Cidades Inteligentes (*Smart Cities*), é notório que a inteligência urbana passa sem dúvida pela capacidade de inovação de um território. Neste processo existem vários atores, desde a indústria, o tecido empresarial, as universidades, as autoridades públicas e também a sociedade civil. Cada um destes atores tem um papel neste puzzle para tornar as nossas cidades mais eficientes, amigas do ambiente e prósperas, aumentando a competitividade das comunidades através da inovação, ao mesmo tempo que melhoram a qualidade de vida dos cidadãos através de um ambiente mais limpo.

Este cenário será aplicado aos alunos da turma B do 9.º ano de escolaridade, do Agrupamento de Escolas de Portela e Moscavide, e será utilizado em contexto interdisciplinar e da flexibilidade curricular com os contributos das disciplinas de Matemática (domínios: Números e Operações; Resolução de problemas, Raciocínio e Comunicação), Físico-Química (domínios: Forças, movimentos e energia), TIC (domínios: Colaboração em ambientes fechados; Exploração de ambientes computacionais), Inglês (domínios: Competência Comunicativa) e Cidadania e Desenvolvimento (domínios: Desenvolvimento Sustentável; Educação Ambiental), este últimos que são domínios obrigatórios a todos os níveis e ciclos de escolaridade, pois são áreas transversais e longitudinais.

O trabalho proposto, contém problemas com uma complexidade crescente (Taxonomia de Bloom). No problema com maior complexidade os alunos serão submetidos a avaliação sumativa, onde devem aplicar a totalidade dos conhecimentos adquiridos na disciplina até então. Quando estamos na presença de um problema complexo, este implica a sua divisão em problemas de menor dimensão, o que estimula a que os alunos se envolvam em dinâmicas de trabalho colaborativo (em equipas), nas quais devem participar ativamente na resolução do problema, utilizando o pensamento computacional e técnicas de programação, nomeadamente na depuração de erros e de sintaxe e lógica.

Pretendo com este cenário, que seja um desafio motivador e que estimule e desencadeie a aprendizagem dos alunos, pois trata-se de um tema atual e de importância acrescida para o presente e futuro. O facto de procurarem soluções através da programação com vista a alcançar a solução dos problemas, com o recurso ao trabalho colaborativo e criativo será uma mais valia para a aprendizagem futura.

Como pré-requisito, os alunos devem ter alguns conhecimentos de programação em linguagens por blocos.

Objetivos de Aprendizagem

Quais os objetivos de aprendizagem assumidos para este cenário? Como se relacionam com as competências que os alunos da turma deverão desenvolver?

- Reconhecer as vantagens do pensamento computacional;
- Reconhecer a vantagem da reutilização do código;
- Reconhecer a função dos blocos de código do programa;
- Identificar os componentes estruturais da programação;
- Identificar os diferentes tipos de dados;
- Identificar os diferentes operadores e funções pré-definidas;
- Identificar o conceito de variáveis;
- Identificar os erros existentes na programação;
- Identificar um problema e decompô-lo em subprogramas (decomposição em partes menores, por semelhança ou redução de complexidade);
- Utilizar a tecnologia de forma segura e responsável;
- Utilizar dos blocos de código do programa;
- Utilizar sensores e atuadores;
- Utilizar as estruturas de seleção e repetição;

- Utilizar as variáveis;
- Resolver os erros existentes na programação e na sequência da resolução do problema;
- Integrar a interdisciplinaridade, favorecendo a integração de conceitos de diversas áreas;
- Elaborar sequências de instruções que envolvam estruturas de seleção, controlo e repetição;
- Elaborar sequências de instruções que envolvam variáveis.

Papel dos Alunos

Em que tipo de atividades serão envolvidos os alunos?

Atividades:

- Revisão dos conceitos básicos de programação;
- Resolução de problemas (construção de soluções para problemas com base nos algoritmos com e sem utilização de sensores/atuadores);
- Implementação dos programas no robot *mBot*, realização de testes e correção de possíveis erros, na procura da solução;
- Apresentação à turma do problema "*Smart City*", discussão das soluções encontradas relativamente à qualidade e eficiência;
- Envio para a plataforma *Google Classroom* da turma das soluções encontradas pelos alunos;

Avaliação:

- Avaliação diagnóstica (questionário de aferição de conhecimentos);
- Avaliação formativa na resolução de problemas diários (listas de verificação por observação direta);
- Avaliação sumativa (resolução de problema final em grupo).

Que tipo de competências Séc. XXI irão essas atividades promover nos alunos?

Este cenário deverá promover a aquisição/desenvolvimento das seguintes competências:

- Relacionamento Interpessoal;
- Desenvolvimento pessoal, da criatividade e autonomia;
- Saber científico, técnico e tecnológico;
- O pensamento crítico e criativo;

- Raciocínio lógico e a capacidade de resolução de problemas;
- Planeamento, trabalho em pares/equipa.

Papel do Professor

Que deve fazer o professor para orientar a aprendizagem e assegurar que os alunos alcancem os seus objetivos?

O Professor deverá ser:

- **Orientador:** promovendo a organização e realização da proposta pedagógica; gerindo os tempos e os grupos de trabalho; proporcionar momentos de avaliação diagnóstica, formativa e sumativa; apoiar a aquisição de conhecimentos por partes dos alunos e promover as suas competências.
- **Incentivador:** promovendo a interação e o enriquecimento das relações interpessoais; prestar ajuda na compreensão e incorporação das tecnologias no decurso da atividade.
- **Inquiridor:** promovendo o pensamento crítico, o pensamento criativo e a reflexão sobre as dificuldades surgidas, através da colocação de questões.
- **Promotor de feedback:** promovendo um feedback construtivo e imediato para permitir ao aluno a correção e orientação, proporcionado assim que o aluno desenvolva a sua autonomia e o seu relacionamento interpessoal.

Que tipo de competências irá estas atividades promover em mim enquanto docente de acordo com o DigCompEdu?

De acordo com o Quadro Europeu de Competência Digital para Educadores (*DigCompEdu*), estas atividades contribuirão para promover as seguintes competências:

Competências profissionais

1. Envolvimento profissional:

1.1. Comunicação institucional: Usar tecnologias digitais para melhorar a comunicação institucional com os aprendentes, encarregados de educação e terceiros. Contribuir, colaborativamente, para desenvolver e melhorar as estratégias de comunicação institucional.

Competências pedagógicas

2. Recursos Digitais:

2.1 Seleção: Identificar, avaliar e selecionar recursos digitais para o ensino e aprendizagem. Ter em consideração o objetivo específico de aprendizagem, o contexto, a abordagem pedagógica e o grupo de aprendentes, ao selecionar recursos digitais e planificar a sua utilização.

2.2 Criação e Modificação: Modificar e desenvolver recursos existentes com licença aberta e outros recursos onde tal é permitido. Criar ou cocriar novos recursos educativos digitais. Ter em consideração o objetivo específico de aprendizagem, o contexto, a abordagem pedagógica e o grupo de aprendentes, ao selecionar recursos digitais e planificar a sua utilização.

3. Ensino e Aprendizagem:

3.1 Ensino: Planificar e implementar dispositivos e recursos digitais no processo de ensino, de modo a melhorar a eficácia das intervenções pedagógicas. Gerir e orquestrar adequadamente estratégias de ensino digital. Experimentar e desenvolver novos formatos e métodos pedagógicos para o ensino.

3.2 Orientação: Usar tecnologias e serviços digitais para melhorar a interação com os aprendentes, individual e coletivamente, dentro e fora da sessão de aprendizagem. Usar tecnologias digitais para proporcionar orientação e assistência oportuna e dirigida. Experimentar e desenvolver novas formas e formatos para oferecer orientação e apoio.

3.3 Aprendizagem colaborativa: Usar tecnologias digitais para promover e melhorar a colaboração do aprendente. Permitir que os aprendentes usem tecnologias digitais enquanto parte de tarefas colaborativas, como meio de melhorar a comunicação, a colaboração e a criação colaborativa de conhecimento.

3.4 Aprendizagem autorregulada: Usar tecnologias digitais para apoiar a aprendizagem autorregulada dos aprendentes, i.e., permitir que planeiem, monitorizem e reflitam sobre a sua própria aprendizagem, forneçam evidências de progresso, partilhem ideias e encontrem soluções criativas.

4. Avaliação:

4.1 Estratégias de avaliação: Usar tecnologias digitais para a avaliação formativa e sumativa. Melhorar a diversidade e adequação dos formatos e abordagens de avaliação.

4.2 Análise de evidências: Produzir, selecionar, analisar criticamente e interpretar evidências digitais sobre a atividade, desempenho e progresso do aprendente, de modo a informar o ensino e aprendizagem.

4.3 Feedback e planificação: Usar tecnologias digitais para fornecer feedback oportuno e direcionado aos aprendentes. Adaptar estratégias de ensino e proporcionar apoio direcionado, com base nas evidências geradas pelas tecnologias digitais utilizadas. Permitir que aprendentes e encarregados de educação compreendam as evidências fornecidas pelas tecnologias digitais e as usem para tomada de decisão.

5. Capacitação dos aprendentes:

5.1 Acessibilidade e inclusão: Garantir acessibilidade a recursos e atividades de aprendizagem para todos os aprendentes, incluindo os que têm necessidades especiais. Ter em consideração e dar resposta às expectativas, capacidades, usos e conceções erróneas (digitais) dos aprendentes, bem como ao uso contextual, físico e cognitivo que fazem das tecnologias digitais.

5.2 Diferenciação e personalização: Usar tecnologias digitais para atender às diversas necessidades de aprendizagem dos aprendentes, permitindo que estes progridam a diferentes níveis e velocidades e sigam caminhos e objetivos de aprendizagem individuais.

5.3 Envolvimento ativo: Usar tecnologias digitais para promover o envolvimento ativo e criativo dos aprendentes com um assunto específico. Usar tecnologias digitais no âmbito de estratégias pedagógicas que fomentem as competências transversais dos aprendentes, a reflexão profunda e a expressão criativa. Abrir a aprendizagem a novos contextos do mundo real, que envolvam os próprios aprendentes em atividades práticas, investigação científica ou resolução de problemas complexos, ou que, de outros modos, aumentem o seu envolvimento ativo em temas complexos.

Ferramentas e Recursos

Que recursos, inclusive tecnológicos, será pertinente usar? De que modo serão usados?

- Computadores portáteis e *tablets* com ligação à Internet;
- Plataforma de gestão de conteúdos escolares *Google Classroom*;
- Promethean *ActivBoard* Interactive Whiteboard (Ecrã Interativo);
- *Robot mBot Explorer Bluetooth V.1*;
- Quadros brancos (lousas pequenas);
- Marcadores para quadro branco;
- Planos de aula e apresentações eletrónicas;

- Guiões de trabalho;
- Tutoriais e textos de apoio;
- *Software* de apoio (*Google Docs, Google Forms, Google Slides, mBlock 5.4.0*).

Pessoas e lugares

Quem mais estará envolvido no cenário (outros docentes, membros da comunidade, empregadores, especialistas externos, etc.) e que papel desempenhará cada um deles? Considere papéis não tradicionais.

Onde terá lugar a aprendizagem: na sala de aula, na biblioteca, ao ar livre, num ambiente online?

Este cenário "*Smart City* será implementado em contexto interdisciplinar e da flexibilidade curricular com os contributos das disciplinas:

- Matemática: A professora vai colaborar na compreensão do funcionamento das variáveis, operadores, vetores;
- Físico-Química: A professora vai colaborar na compreensão dos conceitos associados ao som, luz, velocidade, atrito, potência;
- TIC: O professor vai colaborar na compreensão dos conceitos que serão utilizados neste cenário como utilização de fóruns de discussão, utilização de mensagens instantâneas e salas de conversação, comunicação síncrona/ assíncrona, participação em ambientes colaborativos, adequação do discurso ao contexto de comunicação, pesquisa, análise, gestão e organização da informação na Internet, direitos de autor, utilização de correio eletrónico, segurança e exploração de ambiente computacionais;
- Inglês: A professora vai colaborar na compreensão de todos os conceitos e palavras-chave da linguagem de programação por blocos;
- Cidadania e Desenvolvimento: O professor vai colaborar na compreensão de vários conceitos associados ao desenvolvimento sustentável e à educação ambiental.

Metodologias de Aprendizagem

Que metodologias de aprendizagem e estratégias de ensino serão adotadas? Qual a sua ligação às atividades, aos objetivos e à avaliação?

A metodologia a utilizar será a Aprendizagem Baseada em Problemas (APB), metodologia esta que promove o trabalho em grupo e colaborativo, o pensamento crítico e o pensamento criativo e consequentemente a resolução de problemas.

Para tal ter-se-á em consideração várias etapas:

- Definição do problema;
- Aferição dos conhecimentos prévios dos alunos;
- Identificar os conhecimentos necessários aprender para resolver o problema;
- Investigar sobre o problema e procurar as possíveis soluções;
- Procurar a solução para o problema;
- Avaliar as soluções encontradas para o problema;
- Apresentar resultados (as soluções propostas para o problema).

Tempos

Para a implementação deste cenário estão previstas 10 aulas de 50 minutos, totalizando 500 minutos de intervenção:

Aula n.º	Conteúdos	Duração	Avaliação
0	Ficha de avaliação diagnóstica.	15min	Diagnostica
1	Conceitos básicos de movimento (velocidade, direção) e tempo. Resolução do problema do módulo 1.	50 min	Formativa
2	Conceitos básicos de movimento (velocidade, direção) e tempo. Continuação da resolução do problema da aula anterior.	50 min	Formativa
3	Apresentação do sensor de ultrassom e o sensor seguidor de linha. Resolução do problema do módulo 2.	50 min	Formativa
4	Utilização dos sensores de ultrassom e sensor seguidor de linha. Continuação da resolução do problema da aula anterior introduzindo obstáculos no seu percurso.	50 min	Formativa
5	Apresentação do problema final "Smart City" do módulo 3, com a utilização do sensor de som e módulo LED RGB.	50 min	Formativa
6	Continuação da resolução do problema final "Smart City", com a utilização do módulo LED RGB e o Buzzer.	50 min	Formativa
7	Resolução do problema final "Smart City" do módulo 4, com a utilização do sensor de cor e sensor de luz.	50 min	Formativa
8	Continuação da resolução do problema final "Smart City" do módulo 4, com a utilização do sensor de luz e o módulo LED RGB.	50 min	Formativa
9	Resolução problema final "Smart City" do módulo 5, com a utilização do sensor de movimento PIR (<i>Passive Infrared Sensor</i>).	50 min	Formativa
10	Apresentação e discussão do problema final "Smart City" resolvido pelos grupos de alunos. Autoavaliação. Ficha de avaliação formativa	50 min	Sumativa/Formativa

Aula 0 – Avaliação diagnóstica (no início do mês de fevereiro), antes da aplicação deste cenário.

Aula 1 e 2 – Conceitos básicos de programação e de movimento (velocidade, direção) e tempo (1 problema)

Na primeira e segunda aula procede-se ao registo de presenças, à apresentação do cenário, especificamente os objetivos a atingir pelos alunos e os critérios de avaliação e a constituição

dos grupos de trabalho. De seguida faz-se uma revisão e introdução dos novos conceitos, de seguida apresenta-se um problema (ficha de trabalho), e os alunos resolvem o mesmo (sem a utilização de sensores), assim devem perceber qual será a ação do robot à programação deste, simular o problema, pensar na solução possível, experimentar o comportamento do robot *mBot*, retificar e melhorar, e por fim submeter a solução na plataforma *Classroom* da turma. A aula termina com uma breve síntese dos conteúdos lecionados.

Aula 3 e 4 – Sensores de Ultrassom e Seguidor de Linha (2 problemas)

Na terceira e quarta aula, procede-se ao registo de presenças, revisão da aula anterior, a introdução dos novos conteúdos e a associação das funcionalidades tecnológicas (sensores e atuadores) do robot *mBot* à resolução de problemas específicos, para tal, os alunos devem idealizar a ação do robot, com a utilização dos sensores de ultrassom e de seguidor de linha, retificar e melhorar a programação deste, por último submeter a solução na plataforma *Classroom* da turma. A aula termina com uma breve síntese dos conteúdos lecionados.

Aula 5 e 6 – Módulo LED RGB e o Buzzer (2 problemas)

Na quinta e sexta aula, procede-se ao registo de presenças, revisão da aula anterior, a introdução dos novos conteúdos e a associação das funcionalidades tecnológicas (sensores e atuadores) do robot *mBot* à resolução de problemas específicos, para tal, os alunos devem idealizar a ação do robot, com a utilização do módulo LED RGB e do Buzzer, retificar e melhorar a programação deste, por último submeter a solução na plataforma *Classroom* da turma. A aula termina com uma breve síntese dos conteúdos lecionados.

Aula 7 e 8 – Sensor de luz e o Módulo LED RGB (2 problemas)

Na sétima e oitava aula, procede-se ao registo de presenças, revisão da aula anterior, a introdução dos novos conteúdos e a associação das funcionalidades tecnológicas (sensores e atuadores) do robot *mBot* à resolução de problemas específicos, para tal, os alunos devem idealizar a ação do robot, com a utilização do sensor de luz e o módulo LED RGB, retificar e melhorar a programação deste, por último submeter a solução na plataforma *Classroom* da turma. A aula termina com uma breve síntese dos conteúdos lecionados.

Aula 9 – Sensor de movimento PIR – Passive Infrared Sensor (1 problema)

Na nona aula, procede-se ao registo de presenças, revisão da aula anterior, a introdução dos novos conteúdos e a associação das funcionalidades tecnológicas (sensores e atuadores) do robot *mBot* à resolução de problemas específicos, para tal, os alunos devem idealizar a ação do robot, com a utilização do sensor movimento PIR, retificar e melhorar a programação deste, por último submeter a solução na plataforma *Classroom* da turma. A aula termina com uma breve síntese dos conteúdos lecionados.

Aula 10 – Avaliação Sumativa – Apresentação e discussão dos trabalhos finais.

Na décima aula, procede-se ao registo de presenças, e a apresentação, discussão e avaliação das soluções finais dos problemas de cada grupo de alunos. Finda a apresentação os alunos farão a sua autoavaliação bem como uma ficha de avaliação formativa (na plataforma *Classroom* da turma).

Avaliação

Como as atividades desenvolvidas serão avaliadas (tipo de avaliação, instrumentos, ...)? Sobre o que se foca (objetivos, competências, ...)?

Antes da intervenção (fim de janeiro) será aplicada a avaliação diagnóstica, aferindo assim os conhecimentos prévios dos alunos, por forma a contribuir para a preparação e

operacionalização do cenário e preparar os materiais de acordo com a necessidade específicas de alguns alunos.

A avaliação formativa é efetuada continuamente. Através da lista de verificação de observação direta, regista-se o interesse, a participação nas tarefas, a qualidade de trabalho realizado, a gestão das tarefas, a capacidade de organização.

Assim obtém-se uma informação que permite orientar para que o processo de desenvolvimento das aprendizagens se vá adequando às características dos alunos, permitindo a adaptação do ensino às diferenças individuais. Esta avaliação (com o recurso a rubricas de avaliação) deve ser simples e clara para que seja compreendida pelos alunos, e lhes permita regular a sua aprendizagem, por forma a permitir que cada um construa o seu percurso de aprendizagem.

A avaliação sumativa é efetuada após o trabalho individual e do grupo na resolução dos problemas apresentados, será avaliada a conceção, a realização e apresentação da solução para cada problema, pretende-se que esta avaliação reflita a aprendizagem efetuada ao longo desta intervenção e seja traduzida numa classificação.

A autoavaliação será feita através do preenchimento de um formulário (online), no qual os alunos poderão efetuar a sua própria avaliação (individual).

Narrativa do Cenário de Aprendizagem

Título:

A narrativa do Cenário deve ser redigida para descrever a visão do ensino-aprendizagem da perspetiva do professor ou da perspetiva dos alunos. Considere-a como uma história que descreve a experiência de aprendizagem. Deve ter cerca de 500 palavras e pode descrever uma experiência de aprendizagem tão longa ou tão curta quanto se pretenda, por vezes numa só aula, mas normalmente abrangendo mais do que uma aula, como por exemplo um projeto cuja conclusão possa demorar várias aulas.

Anexo E. Plano de aula n.º 1 e 2

Disciplina: Introdução à Robótica			Data: 23.02.2022 - Horário: 08:55h-09:45h / 09:55h-10:45h		
Unidade Curricular: Introdução à Programação			Aula n.º 1/2		
Ano/Turma: 9.º B			Duração da aula: 100 minutos		
Grupo docência: 550 – Informática			Professores: João Freitas / Paulo Torcato		
Objetivos Gerais: Compreensão dos princípios e conceitos fundamentais de programação, utilizar a Robótica na resolução de problemas. Usar as Tecnologias de Informação e Comunicação (TIC) de forma responsável.			Sumário: Apresentação da intervenção. Revisão de conceitos genéricos sobre algoritmia, robótica e programação por blocos. Conceitos básicos de movimento (velocidade, direção) e tempo. Apresentação do problema a resolver na aula. Entrega da resolução do problema na plataforma <i>Classroom</i> da turma. Síntese da aula.		
Objetivos Específicos	Conteúdos	Atividades	Metodologias / Estratégias	Recursos Técnico / Pedagógicos	Crítérios e Instrumentos de Avaliação
<ul style="list-style-type: none"> - Apresentar os objetivos e temas a desenvolver na intervenção; - Identificar um problema e decompô-lo em subprogramas; - Identificar componentes estruturais da programação; - Compreender e criar instruções que envolvam estruturas de repetição; - Utilizar diferentes meios e aplicações que permitem a comunicação e colaboração em ambientes digitais fechados. 	<ul style="list-style-type: none"> - Componentes de um programa; - Carregamentos do programa do <i>mBlock</i> para o <i>mBot</i>; - Utilização de atuadores de movimento e luz; - Conceitos de movimento (velocidade, direção) e tempo; - Estruturas de controlo e repetição; - Tipos de dados, operadores e expressões lógicas. 	<ul style="list-style-type: none"> - Marcação de presenças; - Apresentação do professor e da intervenção; - Organização dos grupos; - Apresentação do Projeto de Intervenção e respetiva avaliação (ApE); - Apresentação dos conceitos de programação (ApE); - Apresentação do 1.º problema (ApE); - Aceder à plataforma <i>Classroom</i> para o envio do programa; - Esclarecimentos de dúvidas; - Síntese e reflexão me grupo do trabalho desenvolvido. 	<ul style="list-style-type: none"> Expositivo: - Exposição dialogada; Interrogativo: - Colocar questões; Demonstrativo: - Demonstração de procedimentos; Ativo: - Participação na realização dos problemas; Interrogativo: - Debate centrado no conhecimento dos alunos e professores; - Aprendizagem Baseada em Problemas (momentos de reflexão, partilha, discussão, feedback). 	<ul style="list-style-type: none"> - Kit Tecnológico <i>mBot</i> v1; - Computadores e tablets com ligação à Internet; - Quadro interativo; - Canetas; Apresentação eletrónica (ApE); - Tutoriais / Textos de apoio; - Ficha de trabalho (com problemas de complexidade crescente); -Grelhas de observação / Avaliação formativa; - Linguagem de programação por blocos <i>mBlock</i>. 	<ul style="list-style-type: none"> Avaliação Formativa: - Grelha de observação direta.

Anexo F. Plano de aula n.º 3 e 4

Disciplina: Introdução à Robótica			Data: 02.03.2022 - Horário: 08:55h-09:45h / 09:55h-10:45h		
Unidade Curricular: Introdução à Programação			Aula n.º 3/4		
Ano/Turma: 9.º B			Duração da aula: 100 minutos		
Grupo docência: 550 – Informática			Professores: João Freitas / Paulo Torcato		
Objetivos Gerais: Compreensão dos princípios e conceitos fundamentais de programação, utilizar a Robótica na resolução de problemas. Usar as Tecnologias de Informação e Comunicação (TIC) de forma responsável.			Sumário: Revisão da aula anterior. Apresentação do sensor de ultrassom e o sensor seguidor de linha (percurso sem e com obstáculos). Apresentação do problema a resolver na aula. Entrega da resolução do problema na plataforma <i>Classroom</i> da turma. Síntese da aula.		
Objetivos Específicos	Conteúdos	Atividades	Metodologias / Estratégias	Recursos Técnico / Pedagógicos	Crítérios e Instrumentos de Avaliação
<ul style="list-style-type: none"> - Identificar um problema e decompô-lo em subprogramas; - Identificar componentes estruturais da programação; - Compreender e criar instruções que envolvam estruturas de repetição; - Utilizar diferentes meios e aplicações que permitem a comunicação e colaboração em ambientes digitais fechados. 	<ul style="list-style-type: none"> - Componentes de um programa; - Carregamentos do programa do <i>mBlock</i> para o <i>mBot</i>; - Utilização de atuadores de movimento e luz; - Utilização do sensor seguidor de linha; - Estruturas de controlo e repetição; - Tipos de dados, operadores e expressões lógicas. 	<ul style="list-style-type: none"> - Marcação de presenças; - Feedback e apoio aos alunos; - Aceder à plataforma <i>Classroom</i> para o envio do programa; - Esclarecimentos de dúvidas; - Síntese e reflexão me grupo do trabalho desenvolvido. 	<p>Expositivo:</p> <ul style="list-style-type: none"> - Exposição dialogada; <p>Interrogativo:</p> <ul style="list-style-type: none"> - Colocar questões; <p>Demonstrativo:</p> <ul style="list-style-type: none"> - Demonstração de procedimentos; <p>Ativo:</p> <ul style="list-style-type: none"> - Participação na realização dos problemas; <p>Interrogativo:</p> <ul style="list-style-type: none"> - Debate centrado no conhecimento dos alunos e professores; - Aprendizagem Baseada em Problemas (momentos de reflexão, partilha, discussão, feedback). 	<ul style="list-style-type: none"> - Kit Tecnológico <i>mBot</i> v1; - Computadores e tablets com ligação à Internet; - Quadro interativo; - Canetas; - Apresentação eletrónica (ApE); - Tutoriais / Textos de apoio; - Ficha de trabalho (com problemas de complexidade crescente); -Grelhas de observação / Avaliação formativa; - Linguagem de programação por blocos <i>mBlock</i>. 	<p>Avaliação Formativa:</p> <ul style="list-style-type: none"> - Grelha de observação direta.

Anexo G. Plano de aula n.º 5 e 6

Disciplina: Introdução à Robótica			Data: 09.03.2022 - Horário: 08:55h-09:45h / 09:55h-10:45h		
Unidade Curricular: Introdução à Programação			Aula n.º 5/6		
Ano/Turma: 9.º B			Duração da aula: 100 minutos		
Grupo docência: 550 – Informática			Professores: João Freitas / Paulo Torcato		
Objetivos Gerais: Compreensão dos princípios e conceitos fundamentais de programação, utilizar a Robótica na resolução de problemas. Usar as Tecnologias de Informação e Comunicação (TIC) de forma responsável.			Sumário: Revisão da aula anterior. Apresentação do problema final "Smart City", utilizando o módulo LED RGB e o Buzzer. Entrega da resolução do problema na plataforma Classroom da turma. Síntese da aula.		
Objetivos Específicos	Conteúdos	Atividades	Metodologias / Estratégias	Recursos Técnico / Pedagógicos	CrITÉrios e Instrumentos de Avaliação
<ul style="list-style-type: none"> - Identificar um problema e decompô-lo em subprogramas; - Identificar componentes estruturais da programação; - Compreender e criar instruções que envolvam estruturas de repetição; - Utilizar diferentes meios e aplicações que permitem a comunicação e colaboração em ambientes digitais fechados. 	<ul style="list-style-type: none"> - Componentes de um programa; - Carregamentos do programa do mBlock para o mBot; - Utilização de atuadores de movimento e luz; - Utilização do sensor de som, e módulo RGB; - Estruturas de controlo e repetição; - Tipos de dados, operadores e expressões lógicas. 	<ul style="list-style-type: none"> - Marcação de presenças; - Feedback e apoio aos alunos; - Aceder à plataforma Classroom para o envio do programa; - Esclarecimentos de dúvidas; - Síntese e reflexão me grupo do trabalho desenvolvido. 	<ul style="list-style-type: none"> Expositivo: - Exposição dialogada; Interrogativo: - Colocar questões; Demonstrativo: - Demonstração de procedimentos; Ativo: - Participação na realização dos problemas; Interrogativo: - Debate centrado no conhecimento dos alunos e professores; - Aprendizagem Baseada em Problemas (momentos de reflexão, partilha, discussão, feedback). 	<ul style="list-style-type: none"> - Kit Tecnológico mBot v1; - Computadores e tablets com ligação à Internet; - Quadro interativo; - Canetas; Apresentação eletrónica (ApE); - Tutoriais / Textos de apoio; - Ficha de trabalho (com problemas de complexidade crescente); -Grelhas de observação / Avaliação formativa; - Linguagem de programação por blocos mBlock. 	<ul style="list-style-type: none"> Avaliação Formativa: - Grelha de observação direta;

Anexo H. Plano de aula nº 7 e 8

Disciplina: Introdução à Robótica			Data: 16.03.2022 - Horário: 08:55h-09:45h / 09:55h-10:45h		
Unidade Curricular: Introdução à Programação			Aula n.º 7/8		
Ano/Turma: 9.º B			Duração da aula: 100 minutos		
Grupo docência: 550 – Informática			Professores: João Freitas / Paulo Torcato		
Objetivos Gerais: Compreensão dos princípios e conceitos fundamentais de programação, utilizar a Robótica na resolução de problemas. Usar as Tecnologias de Informação e Comunicação (TIC) de forma responsável.			Sumário: Revisão da aula anterior. Continuação da resolução do problema final "Smart City", utilizando o sensor de luz e o módulo LED RGB. Entrega da resolução do problema na plataforma <i>Classroom</i> da turma. Síntese da aula.		
Objetivos Específicos	Conteúdos	Atividades	Metodologias / Estratégias	Recursos Técnico / Pedagógicos	Crítérios e Instrumentos de Avaliação
<ul style="list-style-type: none"> - Identificar um problema e decompô-lo em subprogramas; - Identificar componentes estruturais da programação; - Compreender e criar instruções que envolvam estruturas de repetição; - Utilizar diferentes meios e aplicações que permitem a comunicação e colaboração em ambientes digitais fechados. 	<ul style="list-style-type: none"> - Componentes de um programa; - Carregamentos do programa do <i>mBlock</i> para o <i>mBot</i>; - Utilização de atuadores de movimento e luz; - Utilização de sensores de cor e luz; - Estruturas de controlo e repetição; - Tipos de dados, operadores e expressões lógicas. 	<ul style="list-style-type: none"> - Marcação de presenças; - Feedback e apoio aos alunos; - Aceder à plataforma <i>Classroom</i> para o envio do programa; - Esclarecimentos de dúvidas; - Síntese e reflexão me grupo do trabalho desenvolvido. 	<ul style="list-style-type: none"> Expositivo: - Exposição dialogada; Interrogativo: - Colocar questões; Demonstrativo: - Demonstração de procedimentos; Ativo: - Participação na realização dos problemas; Interrogativo: - Debate centrado no conhecimento dos alunos e professores; - Aprendizagem Baseada em Problemas (momentos de reflexão, partilha, discussão, feedback). 	<ul style="list-style-type: none"> - Kit Tecnológico <i>mBot v1</i>; - Computadores e tablets com ligação à Internet; - Quadro interativo; - Canetas; - Apresentação eletrónica (ApE); - Tutoriais / Textos de apoio; - Ficha de trabalho (com problemas de complexidade crescente); - Grelhas de observação / Avaliação formativa; - Linguagem de programação por blocos <i>mBlock</i>. 	<ul style="list-style-type: none"> Avaliação Formativa: - Grelha de observação direta;

Anexo I. Plano de aula n.º 9 e 10

Disciplina: Introdução à Robótica		Data: 23.03.2022 - Horário: 08:55h-09:45h / 09:55h-10:45h			
Unidade Curricular: Introdução à Programação		Aula n.º 9/10			
Ano/Turma: 9.º B		Duração da aula: 100 minutos			
Grupo docência: 550 – Informática		Professores: João Freitas / Paulo Torcato			
Objetivos Gerais: Compreensão dos princípios e conceitos fundamentais de programação, utilizar a Robótica na resolução de problemas. Usar as Tecnologias de Informação e Comunicação (TIC) de forma responsável.		Sumário: Revisão da aula anterior. Continuação da resolução do problema final "Smart City", utilizando o sensor movimento PIR. Entrega da resolução do problema na plataforma Classroom da turma. Apresentação e discussão dos problemas do cenário de aprendizagem "Smart City" resolvidos pelos grupos de alunos. Autoavaliação. Síntese da aula.			
Objetivos Específicos	Conteúdos	Atividades	Metodologias / Estratégias	Recursos Técnico / Pedagógicos	CrITÉrios e Instrumentos de Avaliação
<ul style="list-style-type: none"> - Identificar um problema e decompô-lo em subprogramas; - Identificar componentes estruturais da programação; - Compreender e criar instruções que envolvam estruturas de repetição; - Utilizar diferentes meios e aplicações que permitem a comunicação e colaboração em ambientes digitais fechados. 	<ul style="list-style-type: none"> - Componentes de um programa; - Carregamentos do programa do <i>mBlock</i> para o <i>mBot</i>; - Utilização de atuadores de movimento e luz; - Estruturas de controlo e repetição; - Utilização de sensor de movimento PIR; - Tipos de dados, operadores e expressões lógicas. 	<ul style="list-style-type: none"> - Marcação de presenças; - Feedback e apoio aos alunos; - Aceder à plataforma Classroom para o envio do programa; - Esclarecimentos de dúvidas; - Síntese e reflexão me grupo do trabalho desenvolvido. 	<p>Expositivo:</p> <ul style="list-style-type: none"> - Exposição dialogada; <p>Interrogativo:</p> <ul style="list-style-type: none"> - Colocar questões; <p>Demonstrativo:</p> <ul style="list-style-type: none"> - Demonstração de procedimentos; <p>Ativo:</p> <ul style="list-style-type: none"> - Participação na realização dos problemas; <p>Interrogativo:</p> <ul style="list-style-type: none"> - Debate centrado no conhecimento dos alunos e professores; - Aprendizagem Baseada em Problemas (momentos de reflexão, partilha, discussão, feedback). 	<ul style="list-style-type: none"> - Kit Tecnológico <i>mBot</i> v1; - Computadores e tablets com ligação à Internet; - Quadro interativo; - Canetas; - Apresentação eletrónica (ApE); - Tutoriais / Textos de apoio; - Ficha de trabalho (com problemas de complexidade crescente); - Grelhas de observação / Avaliação formativa; - Linguagem de programação por blocos <i>mBlock</i>. 	<p>Avaliação Formativa:</p> <ul style="list-style-type: none"> - Grelha de observação direta. <p>Avaliação Sumativa:</p> <ul style="list-style-type: none"> - Grelha de avaliação dos problemas.

Stream **Trabalhos da turma** Pessoas Notas

+ Criar Calendário Google Pasta do Drive da turma

1. Informações

- 1.1 - Estrutura modular da disciplina de Intro... Publicado em 20/02
- 1.2 - Critérios de avaliação (Prática de Ensin... Publicado em 20/02
- 1.3 - Calendarização da Prática Ensino Supe... Editado às 7/05

2. Atividades letivas

2.1 - Sessão n.º 1 - 23 fevereiro

- 2.1.1 - Aula 1/2 (Plano de aula) Publicado em 23/02
- 2.1.2 - Apresentação eletrónica (aula 1 e 2) Publicado em 23/02
- 2.1.3 - Recolha do trabalho de grupo (proble... Publicado em 23/02

2.2 - Sessão n.º 2 - 2 março




- 2.2.1 - Aula 3/4 (Plano de aula) Publicado em 2/03
- 2.2.2 - Apresentação eletrónica (aula 3 e 4) Publicado em 2/03
- 2.2.3 - Recolha do trabalho de grupo (probl... Editado às 15/03

2.3 - Sessão n.º 3 - 16 março

- 2.3.1 - Aula 5/6 (Plano de aula) Publicado em 16/03
- 2.3.2 - Apresentação eletrónica (aula 5 e 6) Publicado em 16/03
- 2.3.3 - Recolha do trabalho de grupo (probl... Editado às 29/03

2.4 - Sessão n.º 4 - 30 março



 2.4.1 - Aula 7/8 (Plano de aula)	Publicado em 30/03
 2.4.2 - Apresentação eletrónica (aula 7 e 8)	Publicado em 30/03
 2.4.3 - Recolha do trabalho de grupo (probl...	Publicado em 30/03

2.5 - Sessão n.º 5 - 4 maio



 2.5.1 - Aula 9/10 (Plano de aula)	Publicado em 4/05
 2.5.2 - Apresentação eletrónica (aula 9 e 10)	Publicado em 4/05
 2.5.3 - Recolha do trabalho de grupo (probl...	Publicado em 4/05

3. Avaliação



 3.1 - Rubrica de avaliação formativa (resolu...	Publicado em 23/02
 3.2 - Inquérito aos alunos sobre intervenção...	Editado às 8/05

4. Recursos



 4.1 - Algoritmos (apresentação eletrónica)	Publicado em 23/02
 4.2 - Manual mBot v1	Publicado em 23/02
 4.3 - Guiões de apoio - mBot - parte 1	Publicado em 23/02
 4.4 - Guiões de apoio - mBot - parte 2	Publicado em 23/02
 4.5 - Vídeos de apoio - Programar com mBot	Publicado em 23/02



mBot
Programação de robots com ...

Aula 1 e 2


23 fevereiro 2022 – 09.00h/10.45h
Professores: João Freitas | Paulo Torcato
9.º B

 INSTITUTO DE
EDUCAÇÃO
DE LISBOA

 Ciências
ULisboa



Objetivos de aprendizagem



- 01 Rever os conceitos de algoritmia
- 02 Rever os conceitos de robótica
- 03 Contruir um robot
- 04 Conhecer o programa mBlock
- 05 Utilizar os blocos de programação
- 06 Identificar e resolver os problemas

Sumário:



- Revisão dos conceitos de: algoritmo, programação e robótica;
- Blocos de Ação: Conceitos básicos de movimento (velocidade, direção) e tempo;
- Resolução do problema do módulo 1.

1. Revisão de conceitos



- i. Revisão do conceito de algoritmo;
- ii. Revisão de conceitos sobre robot *mBot* e seus componentes;
- iii. O mBlock (software de programação por blocos).

Definição:

Um algoritmo pode ser definido como um conjunto de passos finitos e organizados para resolver um determinado problema.

Os algoritmos são procedimentos precisos, não ambíguos, padronizados, eficientes e corretos.

Vídeo:

<https://youtu.be/Da5TOXCwLSg>

<https://www.youtube.com/watch?v=Ct-IOOUqmyY>

Tipo de Algoritmos



- Sandes de queijo
- Trocar lâmpada
- Pesquisa Google
- Pesquisa Maps
- Videoconferência
- Estação espacial

Representação

1. A narrativa descritiva ou descrição narrativa;
2. O fluxograma convencional;
3. O pseudocódigo, também conhecido como linguagem estruturada ou portugal.

Narrativa Descritiva!

Nesta forma de representação os algoritmos são expressos diretamente em linguagem natural.

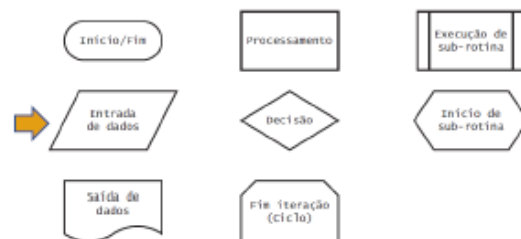
Cálculo da média de um aluno

- Obter as notas da primeira e da segunda prova
- Calcular a média aritmética das provas
- Se a média for maior que 9,5 o aluno é aprovado, senão é reprovado

Fluxograma Convencional!

Um fluxograma é uma representação gráfica de algoritmos onde diferentes formas geométricas implicam distintas ações (instruções, comandos).

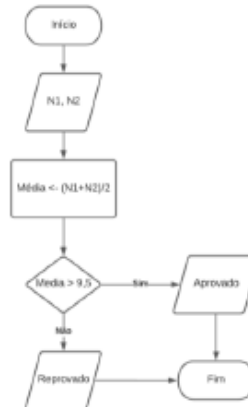
Tal propriedade facilita o entendimento das ideias contidas nos algoritmos.



i. Algoritmos



Representação de um algoritmo de cálculo da média de um aluno sob a forma de um fluxograma convencional.



i. Algoritmos



Pseudocódigo

O pseudocódigo é uma linguagem artificial e informal que ajuda os programadores a desenvolver algoritmos. Esta forma de representação de algoritmos é a mais rica em detalhes.

i. Algoritmos



Representação de um algoritmo de cálculo da média de uma aluno sob a forma de pseudocódigo

(*representação no VisuAlg)



```
Algoritmo Média
Var N1, N2, Média: Real
Início
  Leia N1, N2
  Média <- (N1 + N2)/2
  Se Média > 9,5
    Então
      Escreva "Aprovado"
    Senão
      Escreva "Reprovado"
  Fim_se
Fim
```

ii. Robot *mBot* e componentes



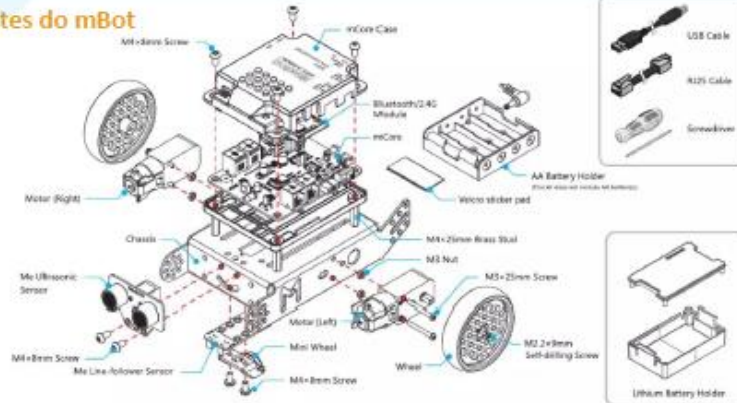
Vem conhecer-me !



ii. Robot mBot e componentes



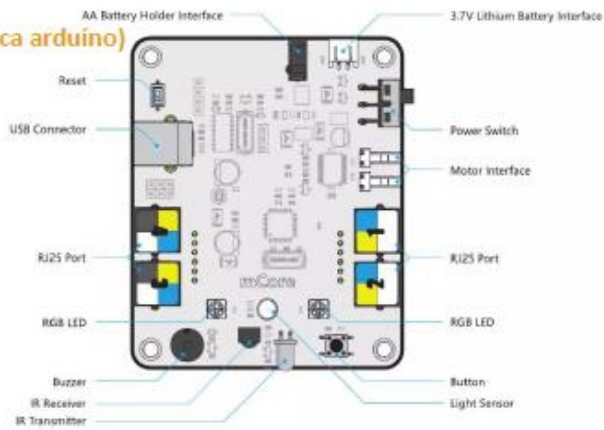
Componentes do mBot



ii. Robot mBot e componentes



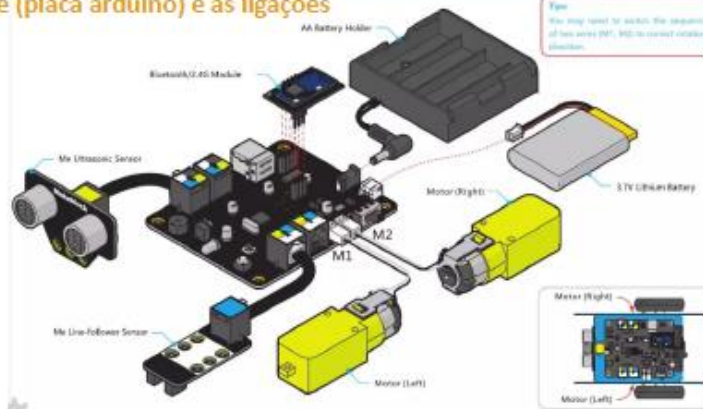
mCore Interface (placa arduino)



ii. Robot *mBot* e componentes



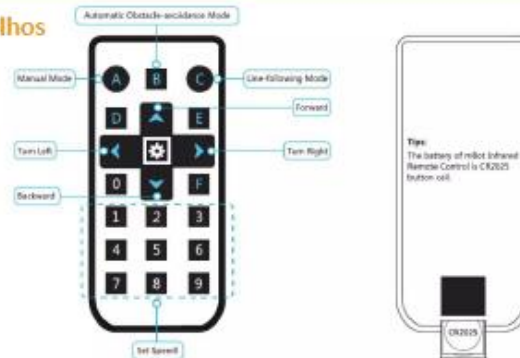
mCore Interface (placa arduino) e as ligações



ii. Robot *mBot* e componentes



Comando de infravermelhos



Tip: If the speed you set is too low, mBot may not move. In this case, just set higher speed.



ii. Robot *mBot* e componentes



	Me 3-Axis Accelerometer and Gyro Sensor provides movement and position - useful for detecting bumps for some robots.			Me Sound Sensor measures volume. It could be used in a project that uses a voice operated switch.	
	Me 7-segment Serial Display shows information such as time, speed or scores to be displayed.			Me Temperature Sensor Module measures temperature, ranging from -55°C to 121°C.	
	Me Potentiometer senses rotation - this information could then be used to control the speed of a robot, the brightness of a light, or anything else you could imagine.			Me Passive IR Detector Module detects infrared signals from humans or animals within 6 meters.	
	Me Joystick can control the movement of a robot or a part of a robot.		Sensores mBot		

iii. Software *mBlock*



mBlock 5



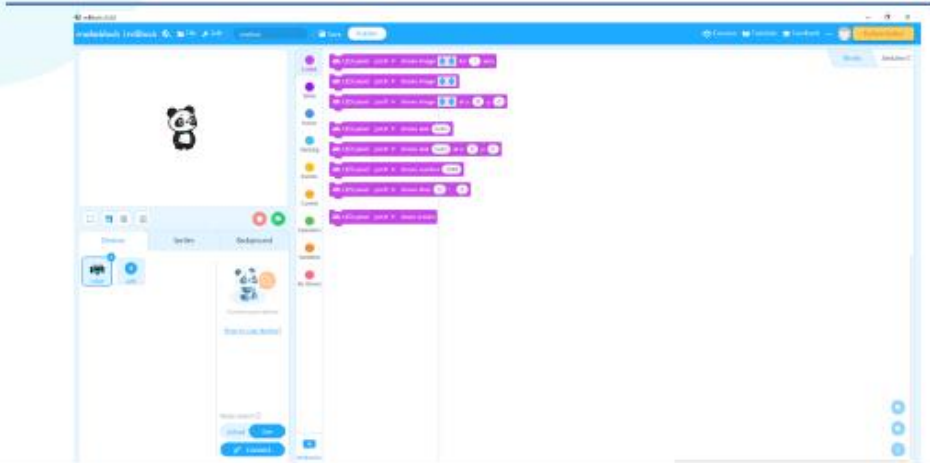
=



+



iii. Software mBlock



iii. Software mBlock



Vem
programar
comigo !



2. Blocos de ação





- i. Blocos de movimento e tempo;
- ii. Blocos de direção e tempo;
- iii. Blocos de direção e paragem

i. Blocos de movimento e tempo



Na categoria "**Action**", podemos encontrar os blocos de movimento e tempo:

 move forward at power 50 % for 1 secs

 move backward at power 50 % for 1 secs

ii. Blocos de direção e tempo



Na categoria **"Action"**, podemos encontrar os blocos de direção e tempo:

turn left at power 50 % for 1 secs

turn right at power 50 % for 1 secs

iii. Blocos de direção e paragem



Na categoria **"Action"**, podemos encontrar os blocos de direção e paragem:

move forward at power 50 %

✓ move forward
move backward
turn left
turn right

left wheel turns at power 50 %, right wheel at power 50 %

stop moving



Resolve o problema com o teu grupo.

Submete o código no Classroom da turma.

Consulta os guíões de ajuda.

Bom trabalho.



Programação de robots com ...





Aula 3 e 4

2 março 2022 – 09.00h/10.45h
Professores: João Freitas | Paulo Torcato
9.º B



Objetivos de aprendizagem



- 01 Rever os conceitos de robótica
- 02 Rever o problema da aula anterior
- 03 Utilizar os blocos de programação
- 04 Descrever como funciona o sensor ultrassónico e o sensor seguidor de linha
- 05 Utilizar blocos de repetição para controlar o robot
- 06 Identificar e resolver os problemas

Sumário:



- Revisão dos conteúdos lecionados na aula anterior;
- Apresentação do sensor de ultrassom e o sensor seguidor de linha (percurso sem e com obstáculos);
- Apresentação do problema a resolver na aula;
- Entrega da resolução do problema na plataforma Classroom da turma;
- Síntese da aula.

1. Revisão da aula anterior



- i. Revisão do conceito de algoritmo;
- ii. Revisão sobre os blocos de código.

i. Algoritmos



Definição:


Um algoritmo pode ser definido como um conjunto de passos finitos e organizados para resolver um determinado problema.

Os algoritmos são procedimentos precisos, não ambíguos, padronizados, eficientes e corretos.

ii. Blocos de movimento e tempo



Na categoria "**Action**", podemos encontrar os blocos de movimento e tempo:

 move forward at power 50 % for 1 secs

 move backward at power 50 % for 1 secs

ii. Blocos de direção e tempo



Na categoria “**Action**”, podemos encontrar os blocos de direção e tempo:

turn left at power 50 % for 1 secs

turn right at power 50 % for 1 secs

ii. Blocos de direção e paragem



Na categoria “**Action**”, podemos encontrar os blocos de direção e paragem:

move forward at power 50 %

- ✓ move forward
- move backward
- turn left
- turn right

left wheel turns at power 50 %, right wheel at power 50 %

stop moving

ii. Revisão do Problema módulo 1



Programa o teu Robot *mBot* para fazer um dos percursos assinalados do local **A** até ao local **B**, utilizando somente os blocos de movimento e direção.



2. Sensores



É um dispositivo que dá um sinal para a deteção ou medição de um elemento físico (calor, luz, som, pressão, magnetismo) ao qual responde.

Este dispositivo recebe um sinal ou um estímulo e responde com um sinal elétrico.



i. Sensor ultrassônico



O sensor ultrassônico é como os olhos do *mBot*, este sensor permite que o *mBot* saiba a distância de um objeto à sua frente.

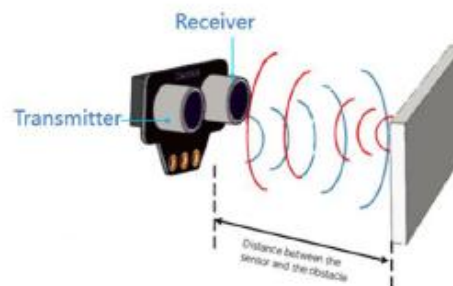
O sensor ultrassônico mede em centímetros (cm) com precisão de 1cm e distância máxima mensurável de 400cm.



i. Sensor ultrassônico



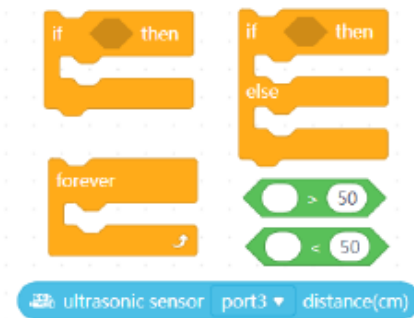
Como funciona o sensor ultrassônico:



i. Sensor ultrassónico



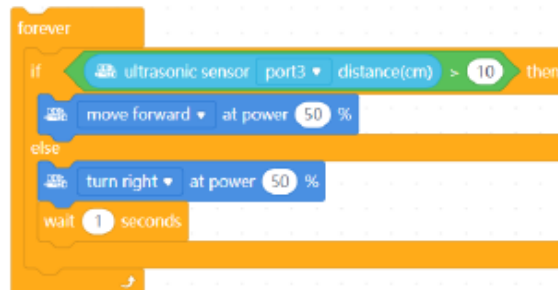
Para a programação deste sensor podemos utilizar alguns blocos das categorias **control**, **operators** e **sensing**.



i. Sensor ultrassónico



Exemplo de um bloco de código para utilização do sensor ultrassónico:



ii. Sensor de seguimento de linha



O sensor de seguimento de linha é o melhor para identificar superfícies brancas ou pretas puras sem reflexão.

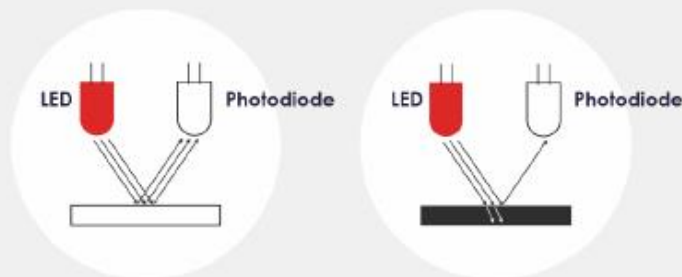
Normalmente, pode-se usar uma fita preta para fazer uma linha de seguimento numa superfície de cor clara para obter os melhores efeitos.



ii. Sensor de seguimento de linha



How the line follower works



ii. Sensor de seguimento de linha



Coding the line follower

line follower sensor port2 value

Sensor 1 (left)	Sensor 2 (right)	Value
Black	Black	0
Black	White	1
White	Black	2
White	White	3

ii. Sensor de seguimento de linha



Situation 1

mBot is on the black line, the value of the line-follower sensor is 0, keeps moving forward

Situation 2

mBot deviates from the black line to the right, the value of the line-follower sensor is 1, should turn left back to find the black line

Situation 3

mBot deviates from the black line to the left, the value of the line-follower sensor is 2, should turn right back to find the black line

Situation 4

mBot is not on the black line, the value of the line-follower sensor is 3, turns back to find the black line

○ Indicate the sensor could not receive the infrared signal

● Indicate the sensor could receive the infrared signal

ii. Sensor de seguimento de linha



Exemplo de um bloco de código para utilização do sensor seguidor de linha:

```
forever
  if line follower sensor port2 value = 0 then
    move forward at power 30 %
  if line follower sensor port2 value = 1 then
    left wheel turns at power 0 %, right wheel at power 30 %
  if line follower sensor port2 value = 2 then
    left wheel turns at power 30 %, right wheel at power 0 %
  if line follower sensor port2 value = 3 then
    move backward at power 30 %
```

3. Problema da aula 3/4



Os congestionamentos referem-se a uma condição em que os automóveis e outros veículo guiam a baixas velocidades com paragens frequentes ou ficam parados em filas, nas estradas, ruas ou avenidas, diminuindo assim o fluxo de movimento.

Este fenômeno também é causado por acidentes ou veículos avariados na via.

3. Problema da aula 3/4



Programa o teu *mBot* para fazer um dos percursos assinalados do local **A** até ao local **B**, utilizando os blocos de movimento, de direção e o sensor ultrassónico para parar quando encontrar um obstáculo.



Resolve o problema com o teu grupo.

Submete o código no Classroom da turma.

Consulta os guiões de ajuda.

Bom trabalho.



Programação de robots com ...



Aula 5 e 6

16 março 2022 – 09.00h/10.45h
Professores: João Freitas | Paulo Torcato
9.º B






INSTITUTO DE
EDUCAÇÃO
—
LISBOA



Ciências
ULisboa



Objetivos de aprendizagem



- 01 Rever os conceitos de robótica
- 02 Rever o problema da aula anterior
- 03 Utilizar os blocos de programação
- 04 Descrever como funcionam os LED's
- 05 Utilizar blocos de programação de LED's e Som (buzzer)
- 06 Identificar e resolver os problemas

Sumário:



- Revisão dos conteúdos lecionados na aula anterior;
- Apresentação dos LED's da placa Arduino do robot, utilização do sensor seguidor de linha (percurso sem e com obstáculos);
- Apresentação do problema a resolver na aula;
- Entrega da resolução do problema na plataforma Classroom da turma;
- Síntese da aula.

1. Revisão da aula anterior



- i. Revisão sobre a utilização dos sensores;
- ii. Revisão sobre os blocos de código.

i. Sensores



É um dispositivo que dá um sinal para a deteção ou medição de um elemento físico (calor, luz, som, pressão, magnetismo) ao qual responde.

Este dispositivo recebe um sinal ou um estímulo e responde com um sinal elétrico.



i. Sensor ultrassónico



O sensor ultrassónico é como os olhos do *mBot*, este sensor permite que o *mBot* saiba a distância de um objeto à sua frente.

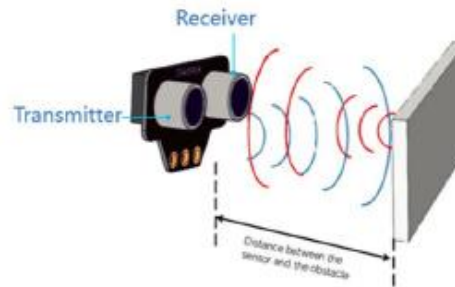
O sensor ultrassónico mede em centímetros (cm) com precisão de 1cm e distância máxima mensurável de 400cm.



i. Sensor ultrassónico



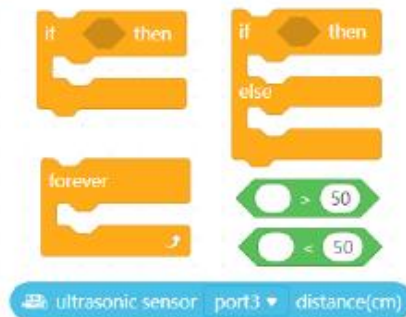
Como funciona o sensor ultrassónico:



i. Sensor ultrassónico



Para a programação deste sensor podemos utilizar alguns blocos das categorias **control**, **operators** e **sensing**.



i. Sensor ultrassónico



Exemplo de um bloco de código para utilização do sensor ultrassónico:

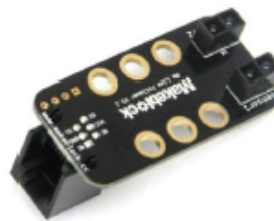
```
forever
  if ultrasonic sensor port3 distance(cm) > 10 then
    move forward at power 50 %
  else
    turn right at power 50 %
  wait 1 seconds
```

ii. Sensor de seguimento de linha



O sensor de seguimento de linha é o melhor para identificar superfícies brancas ou pretas puras sem reflexão.

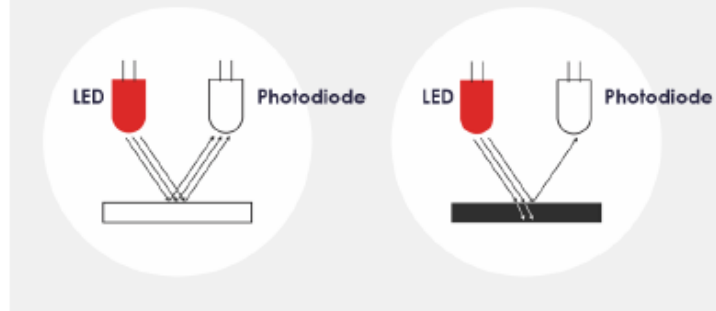
Normalmente, pode-se usar uma fita preta para fazer uma linha de seguimento numa superfície de cor clara para obter os melhores efeitos.



ii. Sensor de seguimento de linha



How the line follower works



ii. Sensor de seguimento de linha

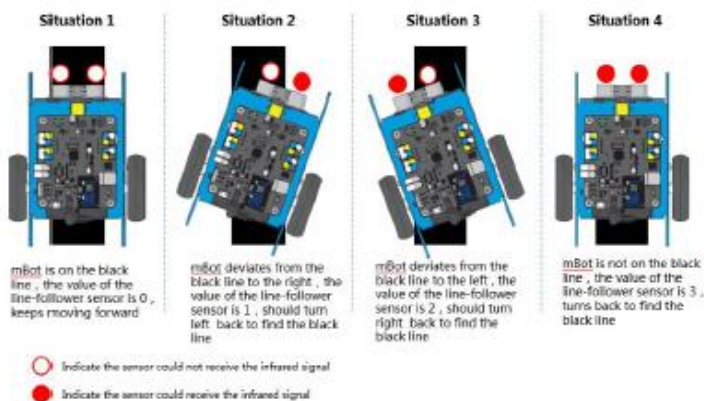


Coding the line follower

line follower sensor port2 value

Sensor 1 (left)	Sensor 2 (right)	Value
Black	Black	0
Black	White	1
White	Black	2
White	White	3

ii. Sensor de seguimento de linha



ii. Sensor de seguimento de linha



Exemplo de um bloco de código para utilização do sensor seguidor de linha:

```
forever
  if line follower sensor - port2 - value = 0 then
    move forward - at power 30 %
  if line follower sensor - port2 - value = 1 then
    left wheel turns at power 0 %, right wheel at power 30 %
  if line follower sensor - port2 - value = 2 then
    left wheel turns at power 30 %, right wheel at power 0 %
  if line follower sensor - port2 - value = 3 then
    move backward - at power 30 %
```

2. RGB LED (light-emitting diode)



A sinalização de emergência (luminosa e sonora), serve para vários propósitos importantes.

O primeiro, é para transmitir de forma visual ou audível uma deslocação urgente ou de emergência.

O segundo, é sinalizar a presença de um veículo de emergência num local de acidente ou um local perigoso.

i. RGB LED



i. RGB LED



Using RGB LEDs

Red	Green	Blue	Result
255	0	0	Red
255	255	0	Yellow
255	255	255	White



ii. BUZZER



play sound at frequency of 700 Hz for 1 secs

play note C4 for 0.25 beats



iii. Código veículo de emergência



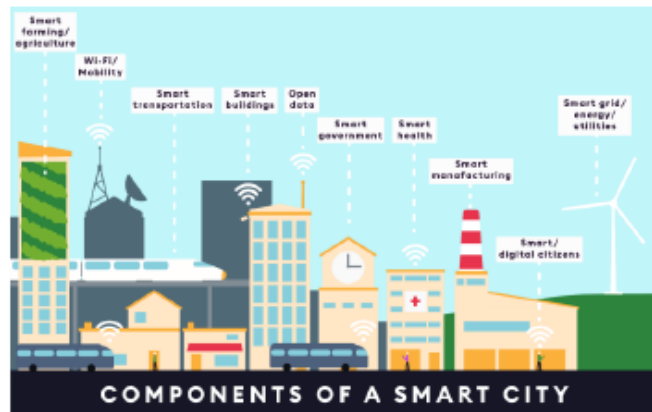
Exemplo de um bloco de código para utilização do LED's e o Buzzer para simulação de um veículo de emergência:

```
when mBot(mcore) starts up
wait until when on-board button pressed ?
move forward at power 30 %
repeat 5
  turn on all light with color red 0 green 0 blue 0
  turn on left light with color red 0 green 0 blue 60
  play sound at frequency of 950 Hz for 0.5 secs
  turn on all light with color red 0 green 0 blue 0
  turn on right light with color red 0 green 0 blue 60
  play sound at frequency of 700 Hz for 0.5 secs
stop moving
```

3. Problema da aula 5/6



As cidades inteligentes (*Smart Cities*) usam informação e comunicação tecnológica para aumentar a sustentabilidade e eficiências dos serviços.



3. Problema da aula 5/6



Programa o teu *mBot* para fazer um dos percursos assinalados do local **A** até ao local **B**, utilizando os blocos de movimento, de direcção e o sensor ultrassónico para parar quando encontrar um obstáculo e ligar o LED e a sirene de emergência.



Resolve o problema com o teu grupo.

Submete o código no Classroom da turma.

Consulta os guíões de ajuda.

Bom trabalho.



Programação de robots com ...





Aula 7 e 8

30 março 2022 – 09.00h/10.45h
Professores: João Freitas | Paulo Torcato
9.º B



Objetivos de aprendizagem



- 01 Rever os conceitos de robótica
- 02 Rever o problema da aula anterior
- 03 Utilizar os blocos de programação
- 04 Utilizar o sensor de luz e módulo LED RGB
- 05 Identificar e resolver os problemas
- 06 Síntese da aula

Sumário:



- Revisão dos conteúdos lecionados na aula anterior;
- Apresentação do sensor de luz;
- Apresentação do problema a resolver na aula;
- Entrega da resolução do problema na plataforma Classroom da turma;
- Síntese da aula.

1. Revisão da aula anterior



- i. Revisão sobre a utilização dos LED's e Buzzer;
- ii. Revisão sobre os blocos de código.

i. RGB LED (light-emitting diode)



A sinalização de emergência (luminosa e sonora), serve para vários propósitos importantes.

O primeiro, é para transmitir de forma visual ou audível uma deslocação urgente ou de emergência.

O segundo, é sinalizar a presença de um veículo de emergência num local de acidente ou um local perigoso.

i. RGB LED



i. RGB LED



Using RGB LEDs

Red	Green	Blue	Result
255	0	0	Red
255	255	0	Yellow
255	255	255	White



i. BUZZER



play sound at frequency of 700 Hz for 1 secs

play note C4 for 0.25 beats



ii. Código veículo de emergência



Exemplo de um bloco de código para utilização do LED's e o Buzzer para simulação de um veículo de emergência:

```
when mBot(micro) starts up
  wait until when on-board button pressed 7
  move forward at power 30 %
  repeat 5
    turn on all light with color red 0 green 0 blue 0
    turn on left light with color red 0 green 0 blue 60
    play sound at frequency of 950 Hz for 0.5 secs
    turn on all light with color red 0 green 0 blue 0
    turn on right light with color red 0 green 0 blue 60
    play sound at frequency of 700 Hz for 0.5 secs
  stop moving
```

2. Módulos: Sensor luz e LED RGB



i. Módulos: Sensor de luz



O sensor de luz permite o controle do robot *mBot* em função da luminosidade. Este permite detetar a intensidade da luz num certo local.



i. Módulo: Sensor de luz



Para a programação deste sensor devemos instalar uma das extensões do *mBlock*, neste caso **Light Sound**.



Light Sound

By mBlock official



Give your mBot lights and sounds effects by building Light-chasing Robot, Scorpion Robot and Smart Lamp.

i. Módulo: Sensor de luz



ii. Módulo LED RGB



Este módulo tem 4 LED's RGB, estes que são possíveis de ser controlados independentemente e ajustar a sua cor e o brilho.



iii. Blocos de código



Exemplo dos blocos de código da extensão do *mBlock: Light Sound*

```
RGB LED port1 lights up all with color red for 1 secs
RGB LED port1 lights up all with color red
RGB LED port1 lights up all with color red 255 green 0 blue 0
 light sensor port3 light intensity
 sound sensor port3 loudness
```

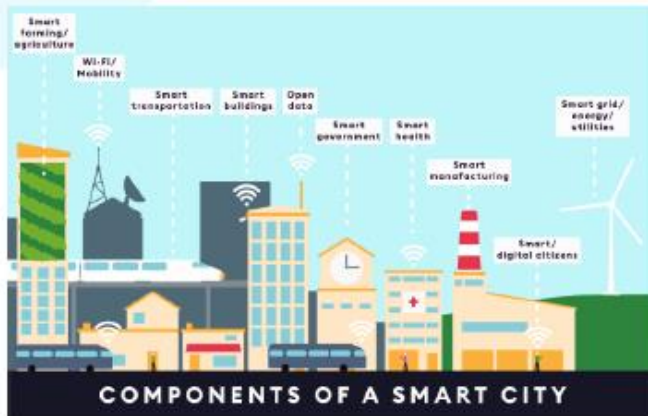
iv. Código sensor luz e LED RGB



Exemplo de um bloco de código para a utilização dos módulos sensor de luz e LED RGB

```
when mBot(mcore) starts up
wait until when on-board button pressed ?
forever
if light sensor port4 light intensity < 500 then
stop moving
play sound at frequency of 700 Hz for 1 secs
RGB LED port1 lights up all with color red for 1 secs
else
move forward at power 50 % for 1 secs
```

3. Problema da aula 7/8



As cidades inteligentes (*Smart Cities*) usam informação e comunicação tecnológica para aumentar a sustentabilidade e eficiências dos serviços.

3. Problema da aula 7/8



Programa o teu *mBot* para fazer um dos percursos assinalados do local **A** até ao local **B**, utilizando os blocos de movimento, de direcção e o módulo sensor de luz para parar quando encontrar ausência de luz e sinalizar com o módulo RGB.





Resolve o problema com o teu grupo.

Submete o código no Classroom da turma.

Consulta os guiões de ajuda.

Bom trabalho.



Programação de robots com ...



Aula 9 e 10

4 maio 2022 – 09.00h/10.45h
Professores: João Freitas | Paulo Torcato
9.º B



INSTITUTO DE
EDUCAÇÃO
—
LISBOA



Ciências
Lisboa



Objetivos de aprendizagem



- 01 Rever os conceitos de robótica
- 02 Rever o problema da aula anterior
- 03 Utilizar os blocos de programação
- 04 Utilizar o sensor de movimento PIR (Passive infrared sensor)
- 05 Identificar e resolver os problemas
- 06 Síntese da aula

Sumário:



- Revisão dos conteúdos lecionados na aula anterior;
- Apresentação do sensor PIR (Passive infrared sensor);
- Apresentação do problema a resolver na aula;
- Entrega da resolução do problema na plataforma Classroom da turma;
- Síntese da aula.

1. Revisão da aula anterior



- i. Revisão sobre a utilização do sensor de luz e do módulo LED RGB;
- ii. Revisão sobre os blocos de código.

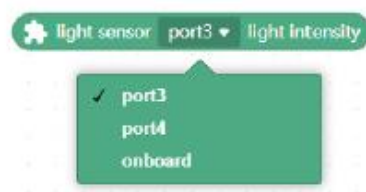
i. Módulos: Sensor luz e LED RGB



i. Módulos: Sensor de luz



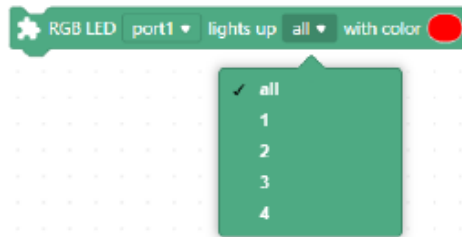
O sensor de luz permite o controle do robot *mBot* em função da luminosidade. Este permite detetar a intensidade da luz num certo local.



i. Módulo LED RGB



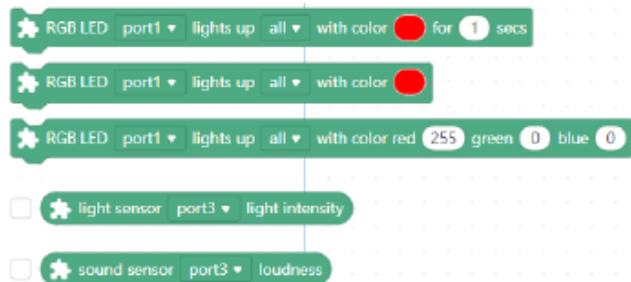
Este módulo tem 4 LED's RGB, estes que são possíveis de ser controlados independentemente e ajustar a sua cor e o brilho.



ii. Blocos de código



Exemplo dos blocos de código da extensão do *mBlock: Light Sound*



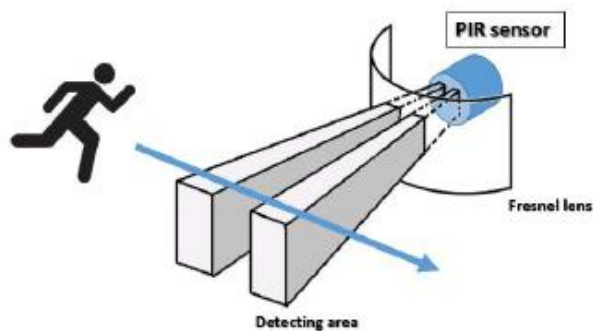
2. Sensor PIR (Passive infrared)



O Módulo Sensor de Movimento PIR é um módulo que deteta a radiação infravermelha emitida pelo corpo humano ou animal, num raio máximo de 6 metros.



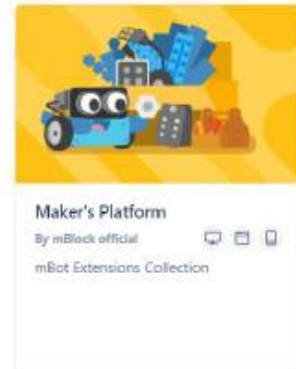
i. Sensor PIR (como funciona)



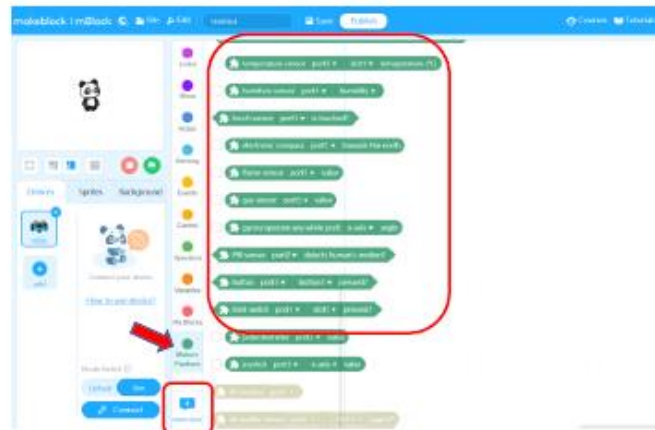
i. Módulo: Sensor de luz



Para a programação deste sensor devemos instalar uma das extensões do *mBlock*, neste caso *Maker's Platform*.



i. Módulo: Sensor de luz



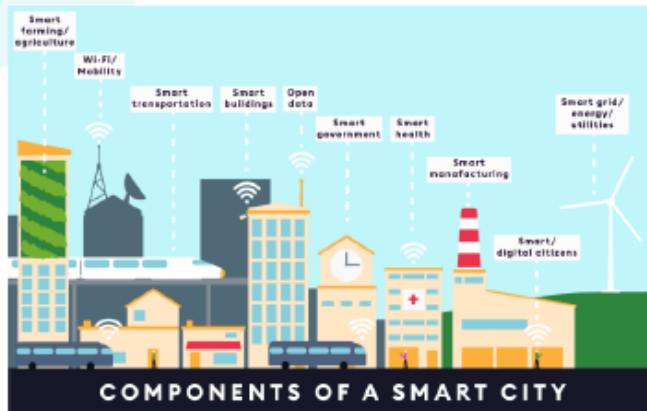
i. Código sensor PIR



Exemplo de um bloco de código para a utilização do sensor PIR (Passive infrared sensor)

```
when mBot(micro) starts up
wait until when on-board button pressed = ?
forever
if PIR sensor port2 detects human's motion? then
  move forward at power 25 % for 2 secs
  play sound at frequency of 700 Hz for 2 secs
  LED all shows color red for 2 secs
else
  move forward at power 50 % for 10 secs
```

3. Problema da aula 9/10

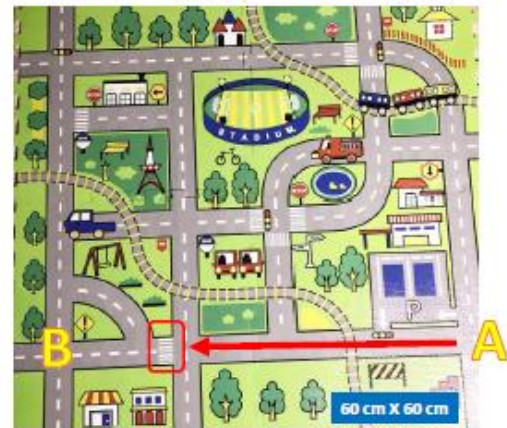


As cidades inteligentes (*Smart Cities*) usam informação e comunicação tecnológica para aumentar a sustentabilidade e eficiências dos serviços.

3. Problema da aula 9/10



Programa o teu *mBot* para fazer um dos percursos assinalados do local **A** até ao local **B**, utilizando os blocos de movimento, e o módulo sensor PIR para o robot abrandar quando detetar movimento humano, na passadeira.



Resolve o problema com o teu grupo.

Submete o código no Classroom da turma.

Consulta os guiões de ajuda.

Bom trabalho.

Escola Secundária da Portela

Sacavém, fevereiro de 2020

Exmo. Senhor Diretor do Agrupamento de Escolas de Portela e Moscavide

Eu, João Paulo Nogueira de Castro Freitas, encontro-me a desenvolver um estudo sobre: “Introdução à Programação e à Robótica Educativa no Ensino Básico através da metodologia de Aprendizagem Baseada em Problemas”. Este estudo é realizado no âmbito da prática de ensino supervisionada, integrada no Mestrado em Ensino da Informática, ministrado pelo Instituto de Educação da Universidade de Lisboa. Este é orientado pela Professora Doutora Neuza Pedro e Professor Doutor Luís Moniz, e sendo ainda o Professor Cooperante Paulo Torcato, docente de disciplina de Introdução à Robótica.

Assim, necessito de fazer a parte empírica deste trabalho que inclui a observação, participação e lecionação das aulas da Turma B do 9.º ano. relativamente a unidade temática de Introdução a Programação, bem como aplicar questionários aos alunos da referida turma.

Solicito então a V. Exa a autorização para proceder ao referido estudo no seu estabelecimento de ensino a Escola Secundaria da Portela.

Comprometo-me a tomar as medidas necessárias, para garantir o normal funcionamento de todo este processo, e também salvaguardar o anonimato dos intervenientes e tratar os dados com confidencialidade.

Sem outro assunto de momento agradeço desde já a sua atenção e disponibilidade e solicito que clique no link abaixo indicado e confirma-se a sua autorização.

Com os meus cumprimentos,

João Freitas

Escola Secundária da Portela

Sacavém, fevereiro de 2020

Exmo(a) Sr(a) Encarregado(a) de Educação

Eu, João Paulo Nogueira de Castro Freitas, encontro-me a desenvolver um estudo sobre: "Introdução à Programação e à Robótica Educativa no Ensino Básico através da metodologia de Aprendizagem Baseada em Problemas". Este estudo é realizado no âmbito da prática de ensino supervisionada, integrada no Mestrado em Ensino da Informática, ministrado pelo Instituto de Educação da Universidade de Lisboa. Este é orientado pela Professora Doutora Neuza Pedro e Professor Doutor Luís Moniz, e sendo ainda o Professor Cooperante Paulo Torcato, docente de disciplina de Introdução à Robótica.

Assim, necessito de fazer a parte empírica deste trabalho que inclui a observação, participação e lecionação das aulas da Turma B do 9.º ano. relativamente a unidade temática de Introdução a Programação, bem como aplicar questionários aos alunos da referida turma.

Solicito a V. Exa a autorização para proceder à recolha de dados e imagens sobre a intervenção atrás descrita comprometendo-me desde já a garantir o anonimato dos alunos e a confidencialidade dos dados obtidos que apenas serão usados como parte empírica do relatório da Prática de Ensino Supervisionada e artigos a publicar em revistas científicas ou a apresentar em congressos nacionais e internacionais.

Sem outro assunto de momento agradeço desde já a sua atenção e disponibilidade e solicito que proceda ao preenchimento do destacável indicado e confirme a sua autorização.

Eu, _____ Encarregado de Educação do(s) aluno(s) _____, n.º ____ da turma B do 9.º Ano, autorizo o(a) meu (minha) educando(a) a participar na recolha de dados e contribuir com a sua participação na Prática de Ensino Supervisionada aluno de Mestrado João Paulo Nogueira de Castro Freitas.

Testes de Normalidade

	Sexo 1 masculino 2 feminino	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
		Estatística	gl	Sig.	Estatística	gl	Sig.
Algoritmia	1	,166	17	,200 [*]	,924	17	,174
	2	,250	8	,150	,897	8	,273
Robótica	1	,206	17	,053	,919	17	,140
	2	,312	8	,021	,840	8	,075
Programação	1	,185	17	,126	,914	17	,117
	2	,443	8	<,001	,601	8	<,001
Pontos	1	,224	17	,023	,901	17	,070
	2	,234	8	,200 [*]	,954	8	,754
Algoritmia	1	,203	17	,060	,915	17	,123
	2	,301	8	,031	,782	8	,018
Robótica	1	,255	17	,004	,848	17	,010
	2	,205	8	,200 [*]	,931	8	,522
Programação	1	,239	17	,011	,888	17	,044
	2	,185	8	,200 [*]	,959	8	,801
Pontos	1	,247	17	,007	,914	17	,117
	2	,151	8	,200 [*]	,938	8	,593

*. Este é um limite inferior da significância verdadeira.

a. Correlação de Significância de Lilliefors

Anexo S. Teste de Homogeneidade de Variância (Estatística de Levene)

Teste de Homogeneidade de Variância

		Estatística de Levene	gl1	gl2	Sig.
Algoritmia	Com base em média	,003	1	23	,957
	Com base em mediana	,029	1	23	,866
	Com base em mediana e com gl ajustado	,029	1	22,017	,866
	Com base em média aparada	,002	1	23	,968
Robótica	Com base em média	,128	1	23	,724
	Com base em mediana	,088	1	23	,770
	Com base em mediana e com gl ajustado	,088	1	19,296	,770
	Com base em média aparada	,164	1	23	,689
Programação	Com base em média	1,601	1	23	,218
	Com base em mediana	2,361	1	23	,138
	Com base em mediana e com gl ajustado	2,361	1	22,956	,138
	Com base em média aparada	1,768	1	23	,197
Pontos	Com base em média	,002	1	23	,963
	Com base em mediana	,036	1	23	,850
	Com base em mediana e com gl ajustado	,036	1	22,850	,850
	Com base em média aparada	,000	1	23	,986
Algoritmia	Com base em média	,039	1	23	,845
	Com base em mediana	,003	1	23	,956
	Com base em mediana e com gl ajustado	,003	1	21,322	,956
	Com base em média aparada	,046	1	23	,832
Robótica	Com base em média	,206	1	23	,654
	Com base em mediana	,113	1	23	,739
	Com base em mediana e com gl ajustado	,113	1	22,698	,740
	Com base em média aparada	,195	1	23	,663
Programação	Com base em média	2,370	1	23	,137
	Com base em mediana	1,588	1	23	,220
	Com base em mediana e com gl ajustado	1,588	1	21,967	,221
	Com base em média aparada	2,310	1	23	,142
Pontos	Com base em média	,219	1	23	,644
	Com base em mediana	,292	1	23	,594
	Com base em mediana e com gl ajustado	,292	1	22,981	,594
	Com base em média aparada	,278	1	23	,603

1. Ficha de avaliação inicial (pré-teste)

Nome: Exemplo

Número: 0

Total de pontos: 0 / 0 (0 %)

1 - Indica a definição correta de algoritmo

Uma sequência de instruções ordenadas, infinitas e não-ambíguas que são empregadas para executar uma tarefa

Uma sequência de instruções desordenadas, finitas e não-ambíguas que são empregadas para executar uma tarefa

→ Uma sequência de instruções ordenadas, finitas e não-ambíguas que são empregadas para executar uma tarefa

5 pontos

2 - Quais as fases de um algoritmo?

Processamento, entrada e saída

→ Entrada, processamento e saída

Entrada, saída e processamento

5 pontos

3 - Quais os estados possíveis que um tipo lógico (booleano) pode assumir?

→ Verdadeiro, Falso

Verdadeiro, Incerto

Falso, Incerto

5 pontos

4 - O que é uma variável?

- Um local na memória do computador usado para armazenar temporariamente dados
- Uma estrutura complexa de cálculo
- Uma estrutura de repetição

5 pontos

5 - Identifica nas opções, quais são os operadores aritméticos.

- + - * / % div
- = <> > < >= <=
- AND, OR, NOT (E, OU, NÃO)

5 pontos

6 - Verifica a imagem com atenção e indica qual o sensor/módulo que está aqui representado:



- Sensor seguidor de linha
- Sensor de luminosidade
- Sensor ultrassónico

5 pontos

7 - Verifica a imagem com atenção e indica qual o sensor/módulo que está aqui representado:



- Sensor seguidor de linha
- Sensor ultrasónico
- Módulo adaptador RJ25 para 6 pinos

5 pontos

8 - Verifica a imagem com atenção e indica qual o sensor/módulo que está aqui representado:



- Sensor de movimento PIR
- Módulo potenciómetro 50k
- Sensor seguidor de linha

5 pontos

9 - Verifica a imagem com atenção e indica qual o sensor/módulo que está aqui representado:



Sensor de movimento PIR

→ Sensor de luz

Sensor seguidor de linha

5 pontos

10 - Verifica a imagem com atenção e indica qual o sensor/módulo que está aqui representado:



→ Módulo joystick

Módulo potenciômetro 50k

Módulo LED RGB

5 pontos

11 - Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que está aqui representado:



- Repete as mensagens
- Repete o conteúdo do ciclo
- Repete uma vez

5 pontos

12 - Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que está aqui representado:



- Se for cumprida a condição então executa o código
- Se estiver vazio então executa o código
- Se for cumprida a condição repete para sempre

5 pontos

13 - Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que está aqui representado:



- Repete as mensagens uma vez
- Repete o conteúdo do ciclo
- Repete o conteúdo do ciclo até cumprir a condição

5 pontos

14 - Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que está aqui representado:



→ Espera até a cumprir a condição

Espera 10 segundos

Espera que o robot se ligue

5 pontos

15 - Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que está aqui representado:



→ Se for cumprida a condição então executa o primeiro bloco de código, senão executa o bloco alternativo

Se for cumprida a condição então executa o segundo bloco de código, senão executa o primeiro bloco de código.

Se estiver vazio então executa só o primeiro bloco de código

5 pontos

16 - Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que está aqui representado:



Liga os LED de 1 a 10 segundos

→ Liga os LED com cores aleatórias de 1 a 10

Liga os LED com a cor verde durante 10 segundos

5 pontos

17 - Verifica a imagem com atenção e indica qual o comportamento do robot através do código aqui representado:



- Repete as instruções do ciclo
- Repete até veres o número 10 no ecrã
- Repete 10 vezes as instruções do ciclo

5 pontos

18 - Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que está aqui representado:



- Move o robot para a frente a 50% de potência após 1 segundo
- Move o robot para a frente a 50% de potência durante 1 segundo
- Move o robot para a frente 50 segundos a 1% de potência

5 pontos

19 - Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que está aqui representado:



- Move a roda esquerda a 30% de potência e a roda direita a 20% de potência
- Move a roda esquerda a 20% de potência e a roda direita a 30% de potência
- Move a roda esquerda a 50% de potência e a roda direita a 50% de potência

5 pontos

20 - Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que está aqui representado:



Liga todos os LED a vermelho

Liga todos os LED durante 1 segundo

→ Liga todos os LED a vermelho durante 1 segundo

5 pontos

Ficha de Avaliação Final (pós-teste)

Nome: Exemplo

Número: 0

Total de pontos: 0 / 0 (0 %)

1 - 1. Indica a definição correta de algoritmo

Uma sequência de instruções ordenadas, infinitas e não-ambíguas que são empregadas para executar uma tarefa

Uma sequência de instruções desordenadas, finitas e não-ambíguas que são empregadas para executar uma tarefa

→ Uma sequência de instruções ordenadas, finitas e não-ambíguas que são empregadas para executar uma tarefa

5 pontos

2 - 2. O que significam respetivamente, os símbolos de fluxogramas aqui representados?



Processo, Decisão, Entrada/Saída

→ Decisão, Processo, Entrada/Saída

Entrada/Saída, Decisão, Processo

5 pontos

3 - 3. O que é uma variável?

→ Um local na memória do computador usado para armazenar temporariamente dados

Uma estrutura complexa de cálculo

Uma estrutura de repetição

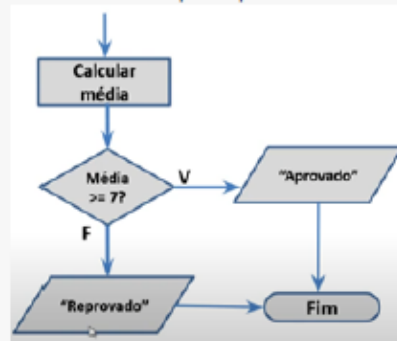
5 pontos

4 - 4. O que entendes por valor de uma constante?

- É um valor que não sofrerá alterações, permanecerá sempre igual, nunca muda
- É um valor que sofrerá alterações, consoante o código
- É um valor que não sofrerá alterações, se estiver dentro de um ciclo de repetição

5 pontos

5 - 5. Se o algoritmo fosse adaptado para as classificações do ensino básico (1 a 5 valores), qual seria a alteração a fazer no bloco de decisão para que o aluno fosse aprovado ou reprovado.



Media = 3

→ Média >= 3

Média <= 3

5 pontos

6 - 6. Verifica a imagem com atenção e indica qual o sensor/módulo que está aqui representado:



- Sensor seguidor de linha
- Sensor de luminosidade
- Sensor ultrassónico

5 pontos

7 - 7. Verifica a imagem com atenção e indica qual o sensor/módulo que está aqui representado:



- Módulo LED RGB
- Módulo adaptador RJ25 para 6 pinos
- Matriz LED RGB

5 pontos

~

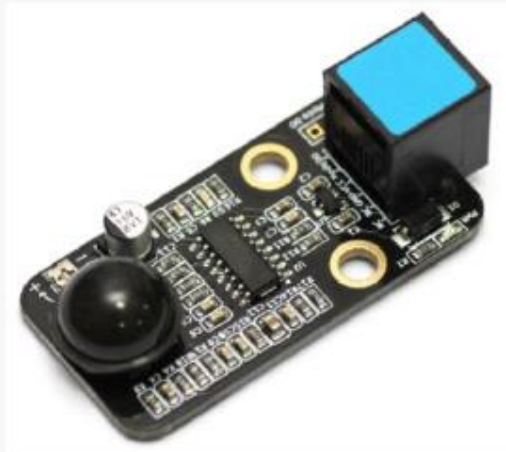
8 - 8. Verifica a imagem com atenção e indica qual o sensor/módulo que está aqui representado:



- Módulo LED RGB
- Módulo potenciómetro 50k
- Matriz LED RGB

5 pontos

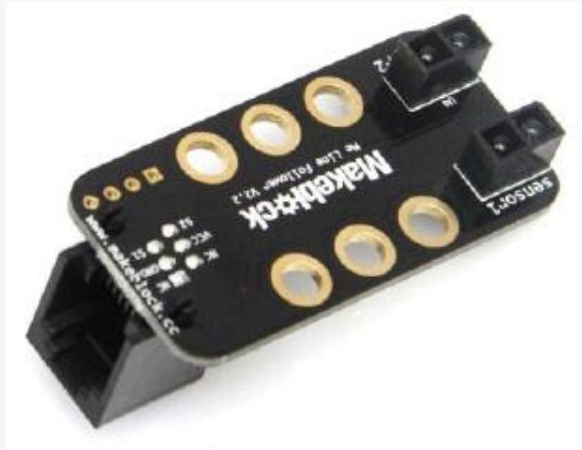
9 - 9. Verifica a imagem com atenção e indica qual o sensor/módulo que está aqui representado:



- Sensor de movimento PIR
- Módulo potenciómetro 50k
- Sensor seguidor de linha

5 pontos

10 - 10. Verifica a imagem com atenção e indica qual o sensor/módulo que está aqui representado:



- Sensor de cor
- Sensor seguidor de linha
- Sensor de movimento PIR

5 pontos

11 - 11. Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que está aqui representado:



- Repete as mensagens
- Repete o conteúdo do ciclo
- Repete uma vez

5 pontos

12 - 12. Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que está aqui representado:



- Se for cumprida a condição então executa o código
- Se estiver vazio então executa o código
- Se for cumprida a condição repete para sempre

5 pontos

13 - 13. Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que está aqui representado:



- Repete as mensagens uma vez
- Repete o conteúdo do ciclo
- Repete o conteúdo do ciclo até cumprir a condição

5 pontos

14 - 14. Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que está aqui representado:

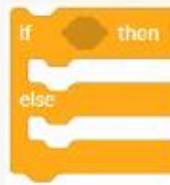


- Espera até a cumprir a condição
- Espera 10 segundos
- Espera que o robot se ligue

5 pontos

15 - 15. Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que

está aqui representado:



→ Se for cumprida a condição então executa o primeiro bloco de código, senão executa o bloco alternativo

Se for cumprida a condição então executa o segundo bloco de código, senão executa o primeiro bloco de código.

Se estiver vazio então executa só o primeiro bloco de código

5 pontos

16 - 16. Verifica a imagem com atenção e indica qual a finalidade do bloco de programação que está aqui representado:



Liga os LED de 1 a 10 segundos

→ Liga os LED com cores aleatórias de 1 a 10

Liga os LED com a cor verde durante 10 segundos

5 pontos

17 - 17. Verifica a imagem com atenção e indica qual o comportamento do robot através do código aqui representado:



O robot anda para a frente á velocidade de 50% durante 5 segundos

O robot anda para a trás à velocidade de 50% durante 5 segundos

→ O robot fica parado

5 pontos

18 - 18. Verifica a imagem com atenção e indica qual o comportamento do robot através do código aqui representado:

```

when mBot(micro) starts up
set a to 10
if a < 10 then
  LED all shows color red for 5 secs
else
  move forward at power 50 % for 5 secs
  
```

O robot liga os LED com a cor vermelha durante 5 segundos

→ O robot anda para a frente à velocidade de 50% durante 5 segundos

O robot liga os LED com a cor vermelha durante 5 segundos e anda para a frente à velocidade de 50% durante 5 segundos

5 pontos

19 - 19. Verifica a imagem com atenção e indica qual o comportamento do robot através do código aqui representado:

```

when mBot(micro) starts up
forever
  move forward at power 50 %
  wait until ultrasonic sensor port3 distance(cm) < 10
  stop moving
  
```

O robot anda em frente durante 10 segundos

→ O robot anda em frente durante e para ao encontrar um objeto a 10 cm

O robot anda em frente e acelera a 50% e ao fim de 10 cm para

5 pontos

20 - 20. Verifica a imagem com atenção e indica qual o comportamento do robot através do código aqui representado:



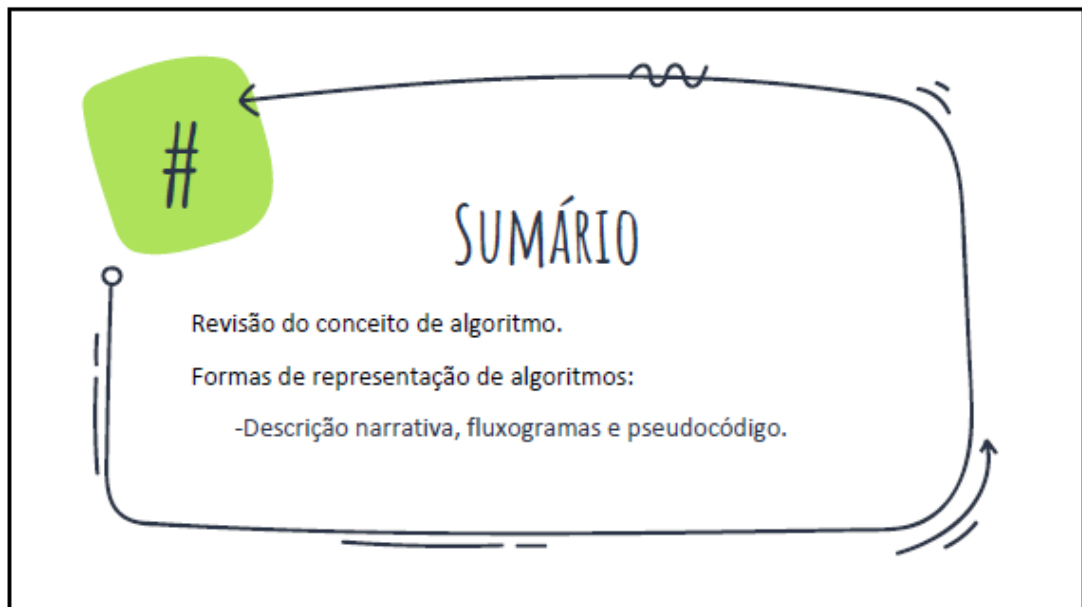
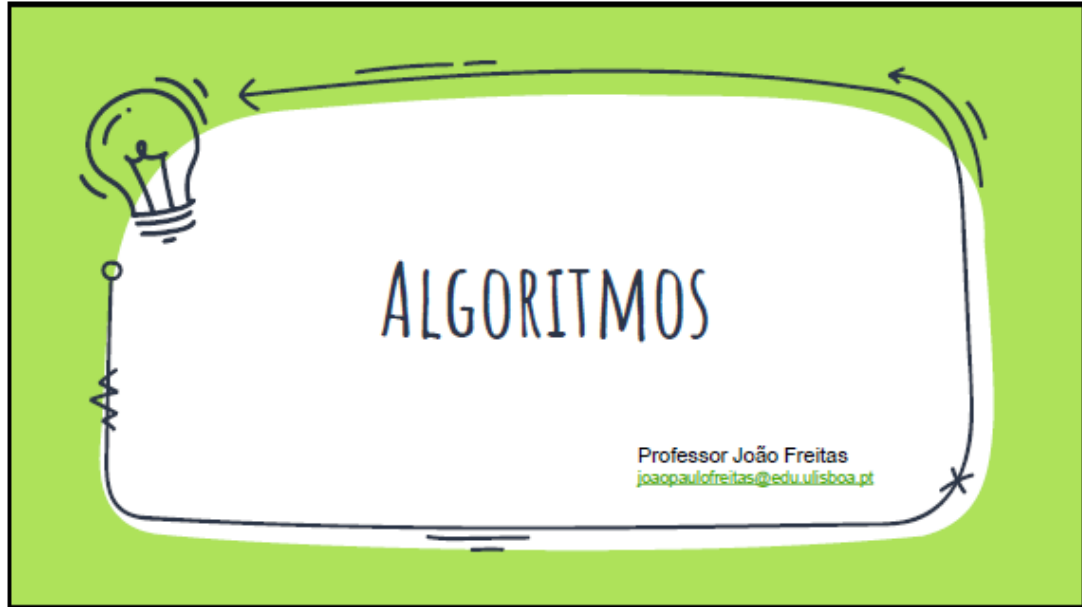
```
when mBot(mcore) starts up
wait until when on-board button pressed ?
set a to 50
set b to 10
move forward at power a % for b secs
```

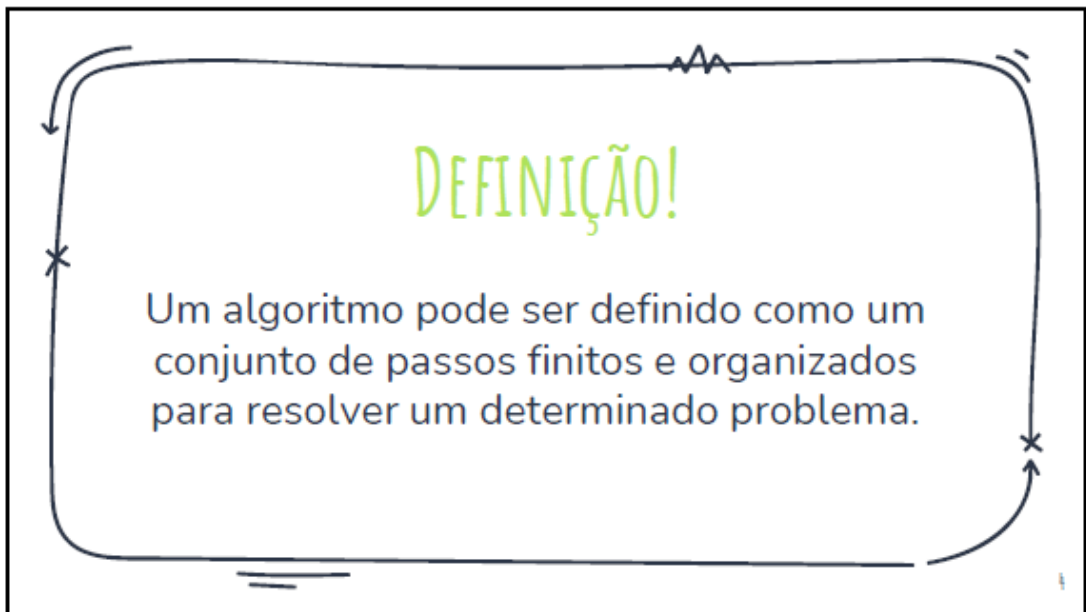
Quando clicar no botão da placa, o robot anda em frente à velocidade de 10% durante 50 segundos

→ Quando clicar no botão da placa, o robot anda em frente à velocidade de 50% durante 10 segundos

Quando clicar no botão da placa, o robot não faz nada

5 pontos





DEFINIÇÃO!

Os algoritmos são procedimentos precisos, não ambíguos, padronizados, eficientes e corretos.

5

O QUE É UM ALGORITMO

VÍDEO:

<https://youtu.be/Da5TOXCwLSg>



6

ALGORITMOS



Sandes de queijo
Trocar lâmpada
Google Search
Google Maps
Videoconferência
Estação espacial

2

REPRESENTAÇÃO DE ALGORITMOS

Formas de representação dos algoritmos

REPRESENTAÇÃO!

1. A narrativa descritiva ou descrição narrativa;
2. O fluxograma convencional;
3. O pseudocódigo, também conhecido como linguagem estruturada ou português.

1

3

NARRATIVA DESCRITIVA

...

NARRATIVA DESCRITIVA!

Nesta forma de representação os algoritmos são expressos diretamente em linguagem natural.

11

NARRATIVA DESCRITIVA – EXEMPLO 1

Receita de bolo

Misture os ingredientes

Unte a forma com
manteiga

Despeje a mistura numa
forma

Se houver coco ralado
então despeje sobre a
mistura

Leve a forma ao forno

Enquanto não corar

deixe a forma no forno

Retire do forno

Deixe arrefecer



12

NARRATIVA DESCRITIVA – EXEMPLO 2

Tomar banho

Entrar na casa de banho e
tirar a roupa

Abrir a torneira do
chuveiro

Entrar na água

Ensaboar-se

Fechar a torneira

Sair da água

Enxugar-se

Vestir-se



13

NARRATIVA DESCRITIVA – EXEMPLO 3

Cálculo da média de um aluno

Obter as notas da primeira e da segunda prova

Calcular a média aritmética das provas

Se a média for maior que 9.5, o aluno é aprovado,
senão é reprovado



14

NARRATIVA DESCRITIVA!

Na prática, esta representação é pouco usada porque o uso da linguagem natural dá muitas vezes oportunidade a más interpretações, ambiguidades e imprecisões.

15

4

FLUXOGRAMA CONVENCIONAL

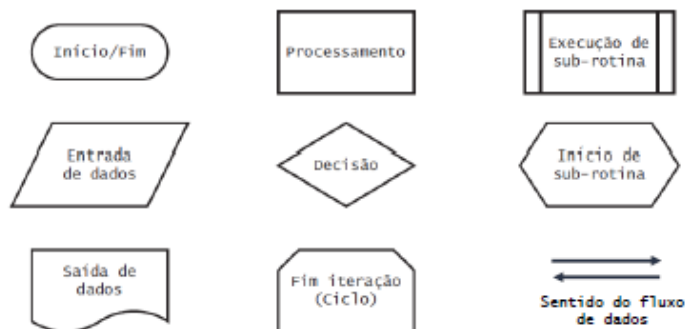
...

FLUXOGRAMA CONVENCIONAL!

Um fluxograma é uma representação gráfica de algoritmos onde diferentes formas geométricas implicam distintas ações (instruções, comandos). Tal propriedade facilita o entendimento das ideias contidas nos algoritmos.

17

Na representação dos fluxogramas, será utilizada a seguinte notação:



18

FLUXOGRAMA CONVENCIONAL!

Um fluxograma tem um símbolo inicial, por onde começa a execução do algoritmo, e um ou mais símbolos finais, que são pontos onde a execução do algoritmo acaba.

19

FLUXOGRAMA CONVENCIONAL!

Partindo do símbolo inicial, existe um caminho orientado para ser seguido, que representa a única sequência de execução das instruções.

20

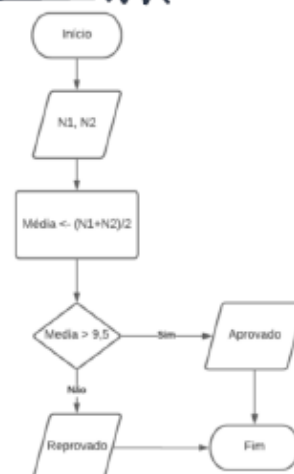
FLUXOGRAMA CONVENCIONAL!

Exceções a esta regra são:

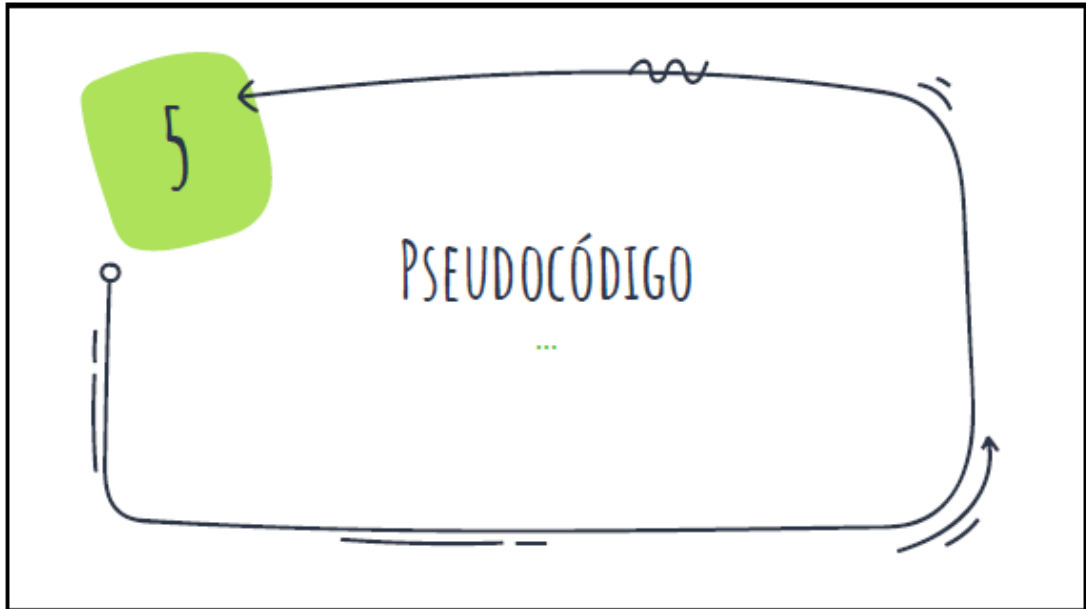
- X os símbolos finais, dos quais não sai nenhum fluxo;
- X os símbolos de decisão, dos quais pode haver mais do que um caminho de saída (normalmente dois caminhos), que representam uma bifurcação no fluxo.

11

Representação de um algoritmo de cálculo da média de um aluno sob a forma de um fluxograma convencional. (*exemplo 3)



12



PSEUDOCÓDIGO

* O pseudocódigo é uma linguagem artificial e informal que ajuda os programadores a desenvolver algoritmos. Esta forma de representação de algoritmos é a mais rica em detalhes.

74

Na escrita de algoritmos em pseudocódigo serão consideradas as seguintes opções:

- X Os algoritmos são delimitados pelas etiquetas Início e Fim;
- X As etiquetas Entrada e Saída são utilizadas na explicação das entradas e saídas de dados;
- X As ações são descritas através de verbos no infinitivo.

25

Na escrita de algoritmos são utilizados os operadores relacionais, lógicos e aritméticos mais comuns, de acordo com a seguinte semântica:

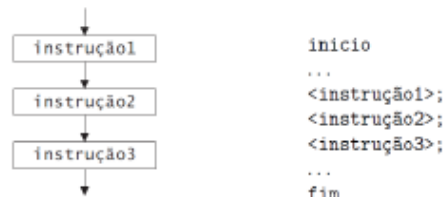
<	menor que
>	maior que
≥	maior ou igual que
≤	menor ou igual que
=	igual
≠	diferente

e, ∧	conjunção
ou, ∨	disjunção
não, ¬	negação

+	soma
-	subtração
*	multiplicação
/	divisão
div	divisão inteira
%	resto da divisão inteira

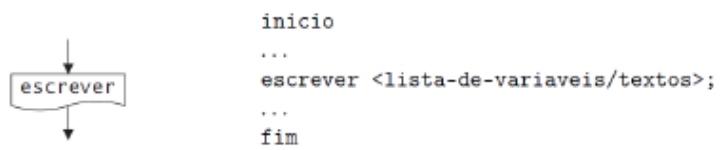
26

As instruções do tipo sequencial são as mais simples de todas, são responsáveis pela entrada/saída de dados, execução de cálculos e atribuição de valores a variáveis:



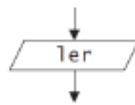
11

As instruções de escrita permitem fazer a saída de dados (tipicamente para o écran) sejam estas variáveis e/ou textos e/ou resultado de cálculos:



11

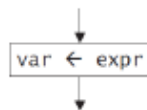
As instruções de leitura permitem fazer a entrada de dados, tipicamente a partir de um teclado, colocando-os em variáveis:



```
inicio
...
ler <lista-de-variaveis>;
...
fim
```

29

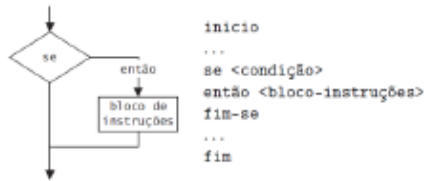
A instrução designada por atribuição permite atribuir o valor de uma expressão a uma variável. A variável que aparece no lado esquerdo da instrução vai assim receber o valor da expressão que aparece no lado direito da mesma instrução:



```
inicio
...
<variável> ← <expressão>;
...
fim
```

30

A decisão binária permite bifurcar a execução de um algoritmo em dois fluxos distintos, para tal é utilizada instrução **Se**. Esta instrução pode ser utilizada de duas formas: **se-então** e **se-então-senão**:



PSEUDOCÓDIGO – EXEMPLO 1

```

Algoritmo <nome_do_algoritmo>
<declaração_de_variáveis>
Início
<corpo_do_algoritmo>
Fim
  
```



PSEUDOCÓDIGO – EXEMPLO 2

Algoritmo é a palavra que indica o início da definição de um algoritmo em forma de pseudocódigo;

<nome_do_algoritmo> é um nome simbólico dado ao algoritmo com a finalidade de distingui-lo dos outros;

<declaração_de_variáveis> é uma forma opcional onde são declaradas as variáveis globais usadas no algoritmo;

Início e Fim são respetivamente as palavras que delimitam o início e o fim do conjunto de instruções do corpo do algoritmo.



33

PSEUDOCÓDIGO – EXEMPLO 3

Representação de um algoritmo de cálculo da média de uma aluno sob a forma de pseudocódigo

(*exemplo 3 - estrutura a ser seguida para representação no VisuAlg)

```
Algoritmo Média
Var N1, N2, Média: Real
Início
  Leia N1, N2
  Média <- (N1 + N2)/2
  Se Média > 9,5
    Então
      Escreva "Aprovado"
    Senão
      Escreva "Reprovado"
  Fim_se
Fim
```



34





Rubrica de avaliação formativa das atividades

Critérios	Fraco (0)	Precisa melhorar (1)	Bom (2)	Muito Bom (3)	Excelente (4)
Processo de resolução do problema	Não colaborou nem se envolveu na resolução do problema.	Colaborou e envolveu-se pouco na resolução do problema.	Colaborou e envolveu-se na resolução do problema.	Colaborou e envolveu-se bastante na resolução do problema.	Colaborou e envolveu-se totalmente na resolução do problema.
Programação do robot	O código não é perceptível, pouca compreensão da utilização dos blocos. Falhas graves na organização e lógica da programação. Tem vários erros.	O código é ligeiramente perceptível, ausência de alguns blocos. Tem alguma organização e lógica da programação. Tem alguns erros.	O código é perceptível, os blocos cumprem quase com o pretendido. Tem organização e lógica da programação. Pode ter alguns erros.	O código é bem perceptível, os blocos cumprem com o pretendido. Tem organização e lógica da programação. Não tem erros.	O código esta bem estruturado e os blocos cumprem com o pretendido. Utiliza funcionalidades avançadas programação. Bem organizado e sem qualquer erro.
Participação no trabalho do grupo	Não colabora com ideias ou comentários acerca do problema.	Colabora pouco com ideias ou comentários acerca do problema.	Colabora com ideias e comentários acerca do problema.	Colabora com bastantes ideias e comentários acerca do problema.	Colabora com clareza e incluem ideias relevantes para a resolução do problema.



Rubrica de avaliação sumativa da atividade final

Critérios	1 Fraco (0 – 19%)	2 Insatisfatório (20 – 49%)	3 Satisfatório (50 – 69%)	4 Bom (70 – 89%)	5 Excelente (90 – 100%)
Processo de resolução do problema (30%)	Não colaborou nem se envolveu na resolução do problema.	Colaborou e envolveu-se pouco na resolução do problema.	Colaborou e envolveu-se na resolução do problema.	Colaborou e envolveu-se bastante na resolução do problema.	Colaborou e envolveu-se totalmente na resolução do problema.
Programação do robot (40%)	O código não é perceptível, pouca compreensão da utilização dos blocos. Falhas graves na organização e lógica da programação. Tem vários erros.	O código é ligeiramente perceptível, ausência de alguns blocos. Tem alguma organização e lógica da programação. Tem alguns erros.	O código é perceptível, os blocos cumprem quase com o pretendido. Tem organização e lógica da programação. Pode ter alguns erros.	O código é bem perceptível, os blocos cumprem com o pretendido. Tem organização e lógica da programação. Não tem erros.	O código está bem estruturado e os blocos cumprem com o pretendido. Utiliza funcionalidades avançadas de programação. Bem organizado e sem qualquer erro.
Participação no trabalho do grupo (20%)	Não colabora com ideias ou comentários acerca do problema.	Colabora pouco com ideias ou comentários acerca do problema.	Colabora com ideias e comentários acerca do problema.	Colabora com bastantes ideias e comentários acerca do problema.	Colabora com clareza e inclui ideias relevantes para a resolução do problema.
Apresentação e defesa do trabalho (10%)	Não participa tanto na apresentação como na defesa do trabalho.	Participa pouco tanto na apresentação como na defesa do trabalho.	Participa tanto na apresentação como na defesa do trabalho.	Participa bastante tanto na apresentação como na defesa do trabalho.	Participa ativamente tanto na apresentação como na defesa do trabalho.

Inquérito de avaliação da Prática de Ensino Supervisionada

Inquérito sobre a intervenção do Professor Estagiário João Freitas, na disciplina de Introdução à Robótica, na turma B do 9.º ano do Agrupamento de Escolas de Portela e Moscavide

*Obrigatório



INSTITUTO DE
EDUCAÇÃO
ULISBOA

1. Nome do aluno (primeiro e último) *

2. Género *

Marcar apenas uma oval.

Masculino

Feminino

1.
Prática
letiva
dos
alunos

Atendendo à tua participação e envolvimento nas atividades letivas, responde as seguintes questões tendo em atenção a seguinte escala:

- 1 - Nunca;
- 2 - Raramente;
- 3 - Algumas vezes;
- 4 - Muitas vezes;
- 5 - Sempre;

3. 1.1. Participaste ativamente na resolução dos problemas propostos? *

Marcar apenas uma oval.

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sempre

4. 1.2. Colaboraste com os teus colegas na resolução dos problemas? *

Marcar apenas uma oval.

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sempre

5. 1.3. Respeitaste os direitos, as opiniões e capacidades dos teus colegas no decurso do trabalho colaborativo em sala de aula? *

Marcar apenas uma oval.

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sempre

6. 1.4. Procuraste definir corretamente a estrutura dos programas elaborados? *

Marcar apenas uma oval.

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sempre

7. 1.5. Escolheste criteriosamente as estruturas de controlo (seleção e/ou repetição) necessárias para resolver os problemas? *

Marcar apenas uma oval.

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sempre

8. 1.6. Verificaste e corrigiste os erros de programação? *

Marcar apenas uma oval.

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sempre

9. 1.7. Reutilizaste o código sempre que foi necessário e apropriado? *

Marcar apenas uma oval.

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sempre

10. 1.8. Demonstraste a compreensão dos programas e dos resultados produzidos? *

Marcar apenas uma oval.

	1	2	3	4	5	
Nunca	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sempre

2. Processo de Ensino e aprendizagem

Atendendo à intervenção do professor e o seu envolvimento na lecionação das atividades letivas, responde as seguintes perguntas tendo em atenção a seguinte escala:

- 1 - Discordo totalmente;
- 2 - Discordo parcialmente;
- 3 - Não concordo nem discordo;
- 4 - Concordo;
- 5 - Concordo totalmente.

11. 2.1. O Professor foi assíduo e pontual? *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

12. 2.2. O Professor apresentou os objetivos de aprendizagem e os problemas de uma forma clara e objetiva? *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

13. 2.3. O Professor revelou conhecimento sobre os conteúdos abordados na intervenção? *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

14. 2.4. O Professor incentivou a participação dos alunos na resolução dos problemas? *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

15. 2.5. O Professor apresentou exemplos de projetos com diferentes graus de dificuldades? *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

16. 2.6. O Professor mostrou-se disponível para o esclarecimento de dúvidas? *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

17. 2.7. O Professor disponibilizou os recursos suficientes na plataforma "Google Classroom" (apresentações eletrônicas, guiões, vídeos, ...)? *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

18. 2.8. As atividades realizadas foram adequadas para a aprendizagem dos conteúdos? *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

19. 2.9. O tempo disponibilizado para cada problema foi suficiente? *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

20. 2.10. A sala reúne as condições para o desenvolvimento da aprendizagem da Robótica Educativa? *

Marcar apenas uma oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

3.
Autoavaliação

Atendendo à tua autoavaliação , responde as seguintes questões tendo em atenção a seguinte escala:

- 1 - Fraco;
- 2 - Insuficiente;
- 3 - Suficiente;
- 4 - Bom;
- 5 - Excelente.

21. 3.1. Foste assíduo? *

Marcar apenas uma oval.

	1	2	3	4	5	
Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Excelente

22. 3.2. Foste pontual? *

Marcar apenas uma oval.

	1	2	3	4	5	
Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Excelente

23. 3.3. Compreendeste os conceitos científicos apresentados? *

Marcar apenas uma oval.

	1	2	3	4	5	
Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Excelente

24. 3.4. Realizaste todas as tarefas propostas nas atividades letivas? *

Marcar apenas uma oval.

	1	2	3	4	5	
Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Excelente

25. 3.5. Estiveste empenhado na realização dos problemas propostos nas atividades letivas? *

Marcar apenas uma oval.

	1	2	3	4	5	
Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Excelente

26. 3.6. Estiveste motivado na realização dos problemas propostos durante as atividades letivas? *

Marcar apenas uma oval.

	1	2	3	4	5	
Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Excelente

27. 3.7. Gostaste de trabalhar em equipa (colaborativamente) na resolução dos problemas propostos? *

Marcar apenas uma oval.

	1	2	3	4	5	
Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Excelente

28. 3.8. Gostaste de trabalhar com a metodologia de Aprendizagem Baseada em Problemas? *

Marcar apenas uma oval.

	1	2	3	4	5	
Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Excelente



Projeto:

Piscar os LEDs do robô MBot

Objetivo:

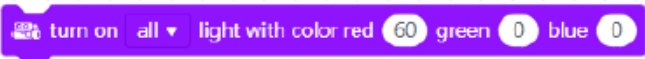
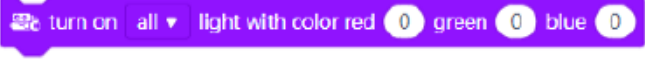

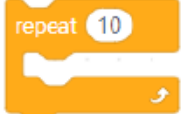
Os Leds internos do robô vão piscar com a cor vermelha durante 1 segundo e depois apagam. Este processo irá ser repetido durante 10 vezes.

Código:

```

when mBot(mcore) starts up
repeat 10
  turn on all light with color red 60 green 0 blue 0
  wait 1 seconds
  turn on all light with color red 0 green 0 blue 0
  wait 1 seconds
  
```

Descrição do código:

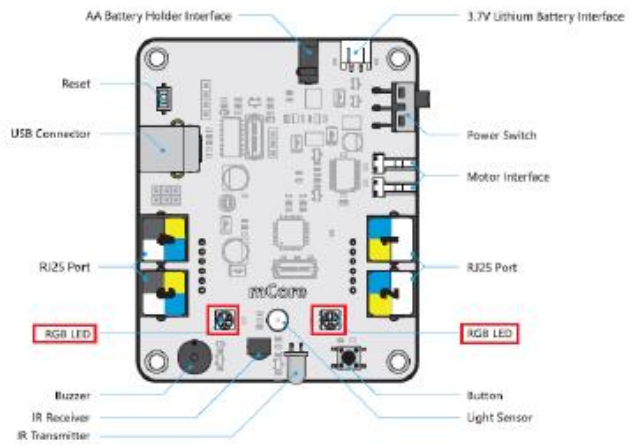
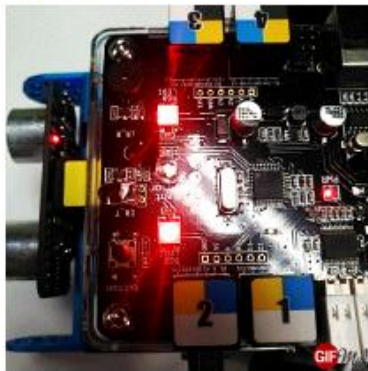
	<p>Os valores definidos para as diferentes cores representam o brilho. Quanto maior for o valor, mais brilho terá o LED. O valor máximo é de 255.</p>
	<p>Quando os parâmetros de cor são 0, a luz LED está desligada.</p>
	<p>Use o bloco "esperar" para controlar o tempo em que o LED fica a vermelho. Este bloco suspende a execução do código por um determinado número de segundos.</p>
	<p>Bloco "repetir" significa repetir o código contido. Este bloco é usado para controlar o número de vezes que o LED pisca.</p>



Desafios:

Tarefa 1	Alterar o script, por forma a manter o LED piscando infinitamente (é necessário fazer o upload do programa para o Arduino).
Tarefa 2	Altera o script, por forma a aumentar a frequência de piscar do LED (é necessário fazer o upload do programa para o Arduino).

Resultado:



Referências:

<http://www.mblock.cc/example/blink/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joapaulofreitas@edu.ulisboa.pt



Projeto:

Piscar os LEDs do mBot de forma alternada

Objetivo:

A luz LED esquerda começa a piscar com cor azul durante 1 segundo e depois desliga-se. De seguida, o LED direito começa a piscar com cor azul durante 1 segundo e depois desliga-se. Esse processo repete-se sem parar.

Código:

```

when mBot(mcore) starts up
repeat 10
  turn on all light with color red 0 green 0 blue 0
  turn on left light with color red 0 green 0 blue 60
  wait 1 seconds
  turn on all light with color red 0 green 0 blue 0
  turn on right light with color red 0 green 0 blue 60
  wait 1 seconds
  
```

Descrição do código:

	<p>Use " todos/led esquerdo/ led direito" para escolher qual o LED. Os valores definidos para as diferentes cores representam o brilho. Quanto maior for o valor, mais brilho terá o LED. O valor máximo é de 255.</p>
	<p>Quando os parâmetros de cor são 0, a luz LED está desligada.</p>
	<p>Serve para controlar a frequência, em termos de segundos, do LED a piscar.</p>

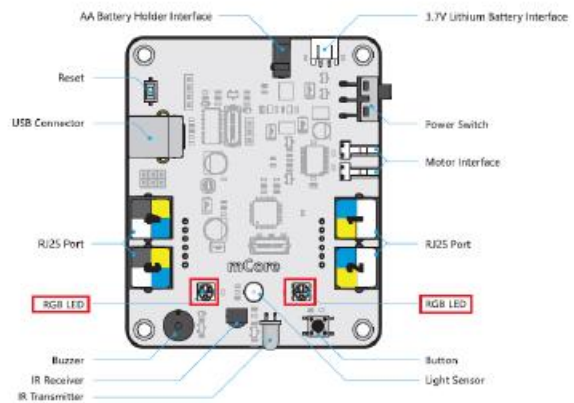
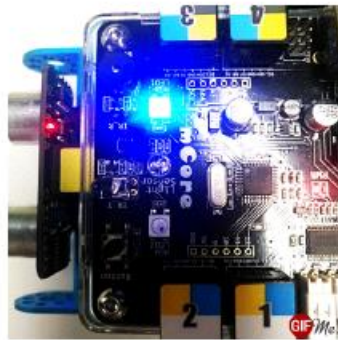


	<p>Bloco que repete continuamente até que o programa seja terminado.</p>
--	--

Desafio:

<p>Tarefa 1</p>	<p>Alterar o código do exemplo para manter a cor da luz do LED a piscar entre vermelho e azul. (é necessário fazer o upload do programa para o Arduino)</p>
-----------------	---

Resultado:



Referências:

<http://www.mblock.cc/example/alternating-blink/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joapaulofreitas@edu.uisboa.pt



Projeto:

Misturar cores do LED RGB

Objetivo:

As luzes do LED da placa mostram vermelho, amarelo, verde, azul, azul, rosa e branco por esta ordem, com 1 segundo de intervalo. Este processo é repetido várias vezes.

Código:

```
when mBot(mcore) starts up
  forever
    turn on all light with color red 60 green 0 blue 0
    wait 1 seconds
    turn on all light with color red 30 green 30 blue 0
    wait 1 seconds
    turn on all light with color red 60 green 0 blue 0
    wait 1 seconds
    turn on all light with color red 0 green 30 blue 30
    wait 1 seconds
    turn on all light with color red 0 green 0 blue 60
    wait 1 seconds
    turn on all light with color red 30 green 0 blue 30
    wait 1 seconds
    turn on all light with color red 20 green 20 blue 20
    wait 1 seconds
```

Descrição do código:

	<p>Ajuste os valores (ou duplo clique na área de número para inserir valores) e deixar o LED da placa misturar a cor vermelho, verde e azul numa certa proporção. No exemplo, as luzes do LED da placa representam as cores vermelho, amarelo, verde, azul, rosa e branco em ordem.</p>
---	---



	Quanto maior o valor, mais brilhante a cor é (intervalo de 0 a 255).
	Serve para controlar a frequência, em termos de segundos, do LED.
	Bloco que repete continuamente até que o programa seja terminado.

Princípios de cor e luz de mistura









A cor do LED da placa resulta da mistura de três cores originais (modelo de cor RGB, ou modelo de cores RGB ou modelo de cor vermelha, verde e azul). É um modelo de agregação de cor, o que poderia mostrar efeitos de cores diversificadas pela mistura de duas ou três cores, em proporções diferentes.



Ele pode ser conhecido a partir do diagrama representado à esquerda que:
 Red+green=yellow
 green+blue=azure
 blue+red=pink
 red+blue+blue=white

RGB Cor Real	Nome	R.G.B	Valor do LED	Código
	Cyan4	0 139 139	0 14 14	
	DodgerBlue	30 144 255	3 14 26	
	SpringGreen2	0 238 118	0 24 12	



	Firebrick	178 34 34	18 3 3	
	VioletRed	208 32 144	21 3 14	
	Maroon	176 48 96	18 5 10	
	DarkGoldenrod 4	139 101 8	14 10 1	

Desafio:

Tarefa 1	Use a tabela de cores RGB para modificar a proporção de luz vermelha, verde e azul para fazer belos efeitos de iluminação. (É necessário fazer o upload do programa para Arduino.)
----------	--

Referências:

<http://www.mblock.cc/example/mixed-color/>



Licença: Creative Commons
 Atribuição não comercial 4.0 Internacional
 (CC BY-NC 4.0)

João Freitas
joapaulofreitas@edu.ulisboa.pt



Projeto:

Cores aleatórias.

Objetivo:

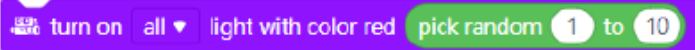
Alterar as cores dos LEDs na placa aleatoriamente a cada 1 segundo, usando números aleatórios.

Código:

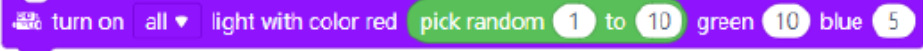
```

when mBot(mcore) starts up
  forever
    turn on all light with color red pick random 1 to 10 green pick random 1 to 10 blue pick random 1 to 10
    wait 1 seconds
  
```

Descrição do código:

	<p>Os intervalos dos números aleatórios são controlados ajustando os seus limites superiores e inferiores.</p>
	<p>Ao substituir o valor da cor com um número aleatório fazemos com que o LED altere a sua cor de forma aleatória.</p>
	<p>A frequência de mudança de cor dos LEDs é controlada pelo tempo de espera.</p>

Desafio:

<p>Tarefa 1</p>	<p>Não alteres os valores verde e azul, escolhe um número aleatório para o vermelho, e observa o efeito de luz.</p>
	
<p>Tarefa 2</p>	<p>Altera o código para permitir que o LED esquerdo e o LED direito da placa mostrem cores diferentes.</p>



```
turn on left light with color red pick random 1 to 10 green pick random 1 to 10 blue pick random 1 to 10  
turn on right light with color red pick random 1 to 10 green pick random 1 to 10 blue pick random 1 to 10
```

Referências:

<http://www.mblock.cc/example/mixed-color/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joaopaulofreitas@edu.ulisboa.pt



Projeto:

Pressione o botão para jogar.

Objetivo:

Quando o botão da placa é pressionado, o alarme soa; quando ele é solto, o alarme não soa.

Código:

```

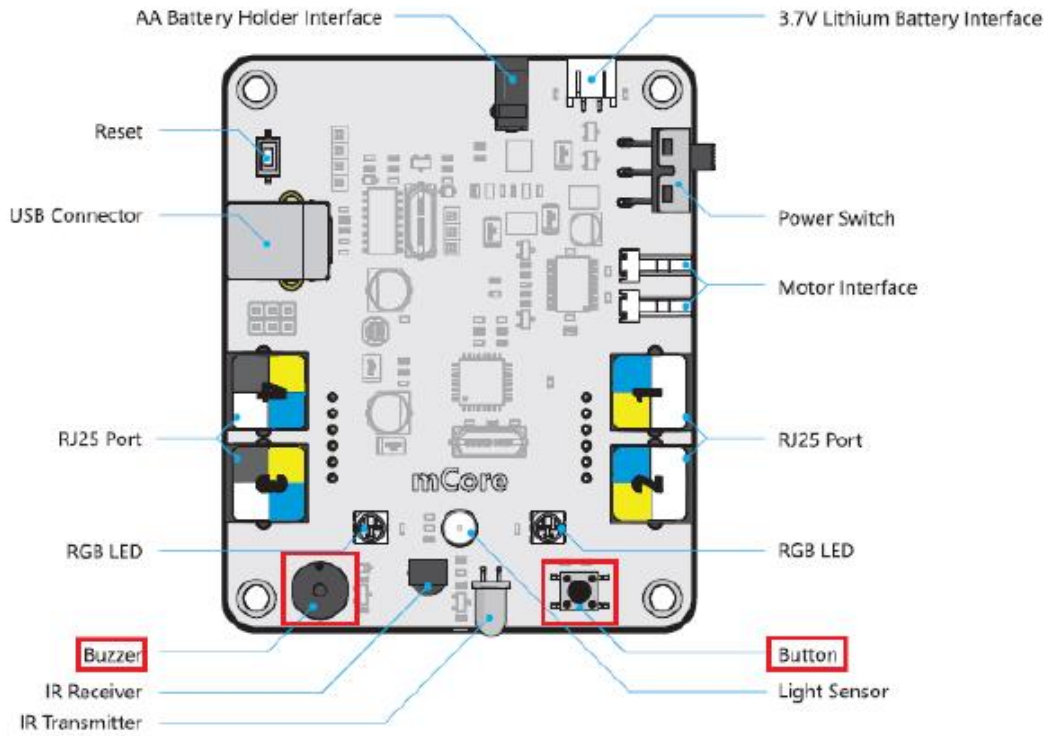
when mBot(mcore) starts up
  forever
    if when on-board button pressed ? then
      play note C4 for 0.25 beats
  
```

Descrição do código:

	<p>Definir o estado do botão da placa</p>
	<p>Definir o tom e ritmos do alarme.</p>
	<p>Quando o programa está em execução, a ação de "tocar uma nota" só é executada se o botão for pressionado.</p> <p>O bloco indica o estado do evento. Se for disparado, o código que se encontra dentro do bloco "Se" é executado; se o evento não é acionado, o código dentro do bloco não é executado. Neste exemplo, quando o botão é pressionado, o código incluído no bloco é executado.</p>



	<p>Repete infinitamente o programa, verificando, constantemente se o botão é pressionado.</p>
---	---



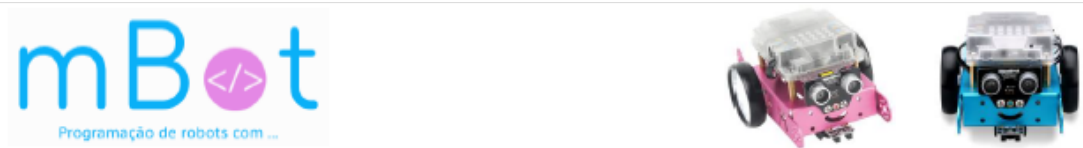
Referências:

<http://www.mblock.cc/example/mixed-color/>



Licença: Creative Commons
 Atribuição não comercial 4.0 Internacional
 (CC BY-NC 4.0)

João Freitas
joaopaulofreitas@edu.ulisboa.pt



Projeto:

Tocar uma peça de música.

Objetivo:





Quando um botão da placa é pressionado, a campainha vai tocar o som Little Star.

Código:

```

when mBot(mcore) starts up
  wait until when on-board button pressed
  play note C4 for 0.25 beats
  play note C4 for 0.25 beats
  play note G4 for 0.25 beats
  play note G4 for 0.25 beats
  play note A4 for 0.25 beats
  play note A4 for 0.25 beats
    
```

Descrição do código:

	<p>Este bloco indica que, se  é disparado, o código que vem a seguir será executado.</p> <p>Se o evento  não é acionado, o restante código não é executado.</p>
	<p>Define o tom e ritmos da campainha.</p>

Os Tons

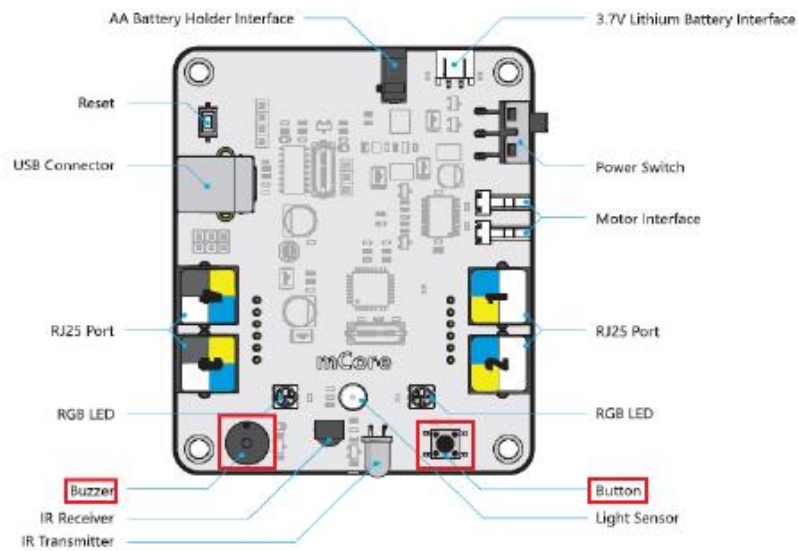
Existe o menu drop-down no bloco que define o tom, o ritmo deve ser colocado manualmente em valores de ¼ ou seja 0.25.

Os valores C / D / E / F / G / A / B no menu tom, definem o nome do tom, por forma a fazer o mapeamento para Do/Re/Mi/Fa/Sol/La/Ti.



Desafios:

Tarefa 1	Consulte a música Little Star e completa os outros segmentos da música.																						
	<table border="1"> <thead> <tr> <th>musical alphabet</th> <th>sofêge</th> <th>numbered musical notation</th> </tr> </thead> <tbody> <tr> <td>CCGGAAG-</td> <td>Do Do So So La La So --</td> <td>1155665-</td> </tr> <tr> <td>FFEEDDC-</td> <td>Fa Fa MI MI Re Re Do --</td> <td>4433221-</td> </tr> <tr> <td>GGFFEED-</td> <td>So So Fa Fa MI MI Re --</td> <td>5544332-</td> </tr> <tr> <td>GGFFEED-</td> <td>So So Fa Fa MI MI Re --</td> <td>5544332-</td> </tr> <tr> <td>CCGGAAG-</td> <td>Do Do So So La La So --</td> <td>1155665-</td> </tr> <tr> <td>FFEEDDC-</td> <td>Fa Fa MI MI Re Re Do --</td> <td>4433221-</td> </tr> </tbody> </table>	musical alphabet	sofêge	numbered musical notation	CCGGAAG-	Do Do So So La La So --	1155665-	FFEEDDC-	Fa Fa MI MI Re Re Do --	4433221-	GGFFEED-	So So Fa Fa MI MI Re --	5544332-	GGFFEED-	So So Fa Fa MI MI Re --	5544332-	CCGGAAG-	Do Do So So La La So --	1155665-	FFEEDDC-	Fa Fa MI MI Re Re Do --	4433221-	
musical alphabet	sofêge	numbered musical notation																					
CCGGAAG-	Do Do So So La La So --	1155665-																					
FFEEDDC-	Fa Fa MI MI Re Re Do --	4433221-																					
GGFFEED-	So So Fa Fa MI MI Re --	5544332-																					
GGFFEED-	So So Fa Fa MI MI Re --	5544332-																					
CCGGAAG-	Do Do So So La La So --	1155665-																					
FFEEDDC-	Fa Fa MI MI Re Re Do --	4433221-																					
Tarefa 2	Consulte outras notações de músicas e escreva mais programas de música.																						



Referências:

<http://www.mblock.cc/example/play-a-piece-of-music/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joaopaulofreitas@edu.uisboa.pt



Projeto:

Som de ambulância.

Objetivo:

Cada vez que pressionarmos o botão da placa, a campainha irá tocar um som de ambulância.

Código:

```

when mBot(mcore) starts up
  forever
    if when on-board button pressed ? then
      repeat 5
        play sound at frequency of 950 Hz for 1 secs
        play sound at frequency of 700 Hz for 1 secs
  
```

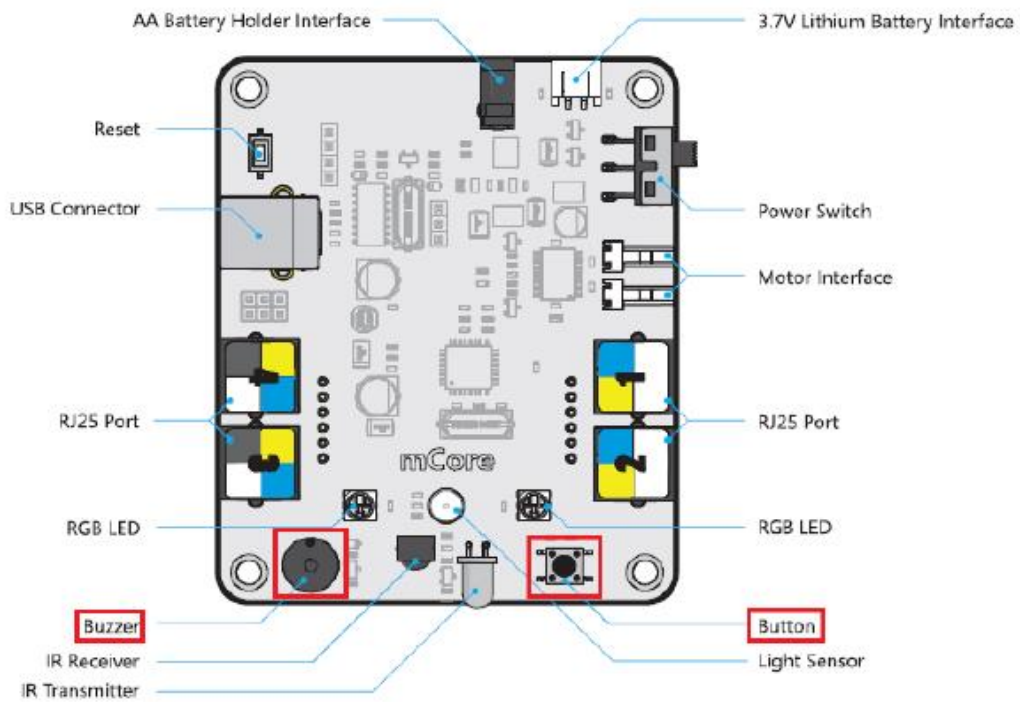
Descrição do código:

	<p>Utilize este bloco para determinar se devemos reproduzir os efeitos sonoros de acordo com o estado do botão da placa.</p>
	<p>Devemos repetir as operações para garantir que os efeitos sonoros são reproduzidos cada vez que o botão é pressionado.</p>
	<p>Definir diretamente o som do alarme sonoro de acordo com a frequência, durante um tempo definido.</p>

Desafios:



Tarefa 1	Pensa se é possível substituir  por  neste exemplo?
----------	--



Referências:

<http://www.mblock.cc/example/ambulance-sound/>



Licença: Creative Commons
 Atribuição não comercial 4.0 Internacional
 (CC BY-NC 4.0)

João Freitas
joapaulofreitas@edu.ulisboa.pt



Projeto:

Gradiente de Som.

Objetivo:

A campainha da placa irá sofrer um efeito crescente no tom do som.

Código:

```

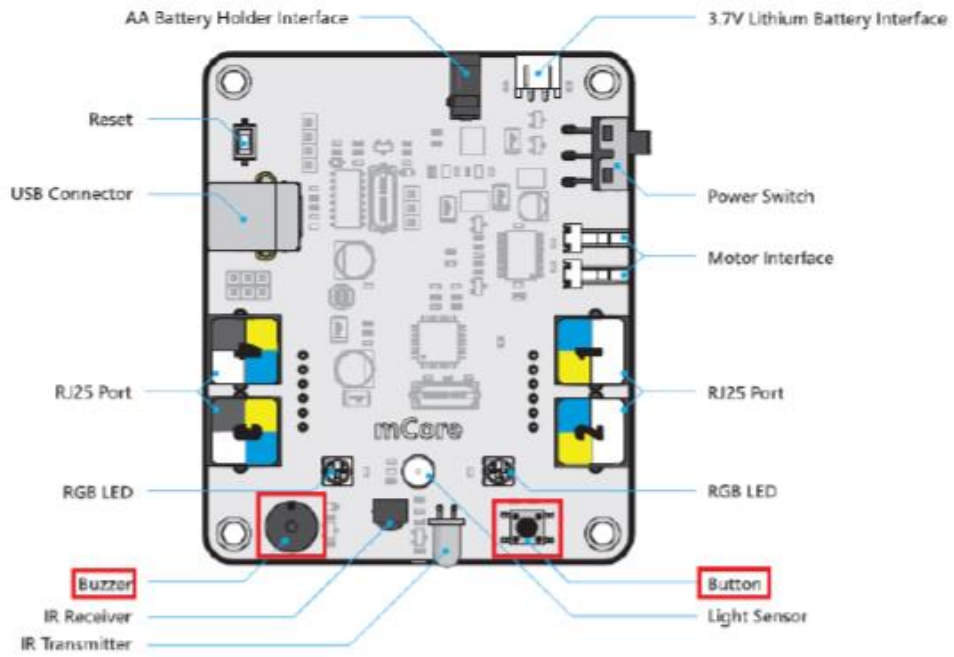
when mBot(mcore) starts up
  set f to 500
  repeat 50
    play sound at frequency of f Hz for 0.25 secs
    change f by 20
  
```

Descrição do código:

	Definir variável f para um tom de frequência e definir a duração de 50ms.
	Definir o valor inicial da variável f em 500.
	Alterar o valor da variável f , incrementando o seu valor em 20.
	Definir o número de vezes que a nota será tocada e que variável f será alterada.

Desafios:

Tarefa 1	Deixe a campainha tocar um som de frequência decrescente.
----------	---



Referências:

<http://www.mblock.cc/example/sound-gradient/>



Licença: Creative Commons
 Atribuição não comercial 4.0 Internacional
 (CC BY-NC 4.0)

João Freitas
joapaulofreitas@edu.ulisboa.pt



Projeto:

Gradiente de Brilho.

Objetivo:

A luz do LED da placa, na cor vermelha, aumenta o brilho gradualmente. Quando o brilho atinge um determinado valor, o LED desliga-se. Repita este processo.

Código:

```
when mBot(mcore) starts up
  forever
    set red to 0
    repeat 60
      turn on all light with color red red green 0 blue 0
      change red by 1
      wait 0.02 seconds
```

Descrição do código:

	Definir variável red como o valor do brilho da luz vermelha do LED.
	Definir o valor inicial da variável red como 0, ficando o LED inicialmente desligado.
	Incrementa o valor da variável red em um.
	Evitar uma mudança excessivamente rápida da luz vermelha. Esperar um pouco para controlar a velocidade de alteração.



	<p>Pode-se ajustar o número "60" aqui defino, acabando por ajustar o brilho máximo que o LED pode alcançar.</p>
--	---

Referências:

<http://www.mblock.cc/example/sound-gradient/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joapaulofreitas@edu.ulisboa.pt



Projeto:

Simular o som dos bombeiros.

Objetivo:

A campanha deve simular os efeitos de som de carros de bombeiros.

Código:

```
when mBot(mcore) starts up
  forever
    change f by 700
    repeat until f > 1500
      play sound at frequency of f Hz for 0.5 secs
      change f by 50
    change f by 1500
    repeat until f < 700
      play sound at frequency of f Hz for 0.5 secs
      change f by -50
```

Descrição do código:

O som de baixa frequência do motor da sirene de incêndio é entre 650Hz e 750Hz, e o seu som de alta frequência é entre 1450Hz e 1550Hz. O som da sirene é gerado através do seguinte padrão: o som de baixa frequência amplifica para um som de alta frequência em 1,5 segundos, e, em seguida, cai de volta para a frequência inferior em 3,5 segundos. Portanto, o som da sirene do carro dos bombeiros pode ser programaticamente simulado da seguinte forma:

Definir a baixa frequência como 700Hz, e, em seguida, definir o som de alta frequência como 1500Hz, e de forma repetida tocar a campainha na faixa de 700Hz a 1500Hz e depois voltar para 700Hz. A razão entre o tempo de amplificação para o tempo de queda é de 1,5: 3,5, que é de 3: 7, de modo que a razão de amplificação de frequência para a queda necessita de ser de 7: 3.

	<p>Definir variável f como a frequência de um tom e definir a duração soando como 12ms (12 aqui é apenas para exemplo, podemos configurá-lo para qualquer outro valor).</p>
--	--



	<p>Definir o limite inferior da variável f como 700 e o limite superior da frequência como 1500.</p>
	<p>Este bloco indica que o código será executado sempre que a condição definida em não seja válida.</p>
	<p>Este bloco significa: definir o valor inicial da frequência de tom como 700Hz e tocar o som durante 0.5 segundos, de seguida a frequência aumenta em 50Hz, até um limite de 1.500 Hz, que quando atingido, quebra o ciclo e a frequência não será mais aumentada.</p>
	<p>Este bloco significa: definir o valor inicial da frequência de tom como 1500Hz e tocar o som durante 0.5 segundos, de seguida a frequência diminui em 50Hz, até um limite de 700 Hz, que quando atingido, quebra o ciclo e a frequência não será mais diminuída.</p>

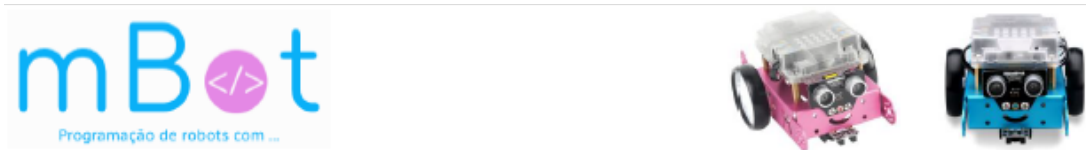
Referências:

<http://www.mblock.cc/example/fire-engine-sound-effects/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joaopaulofreitas@edu.ulisboa.pt



Projeto:

Valor do sensor de luz.

Objetivo:

Quando a luz ambiente em redor do sensor de luz é alterada, o programa irá mostrar, em tempo real, as leituras do sensor de luz da placa na matriz LED.

Código:

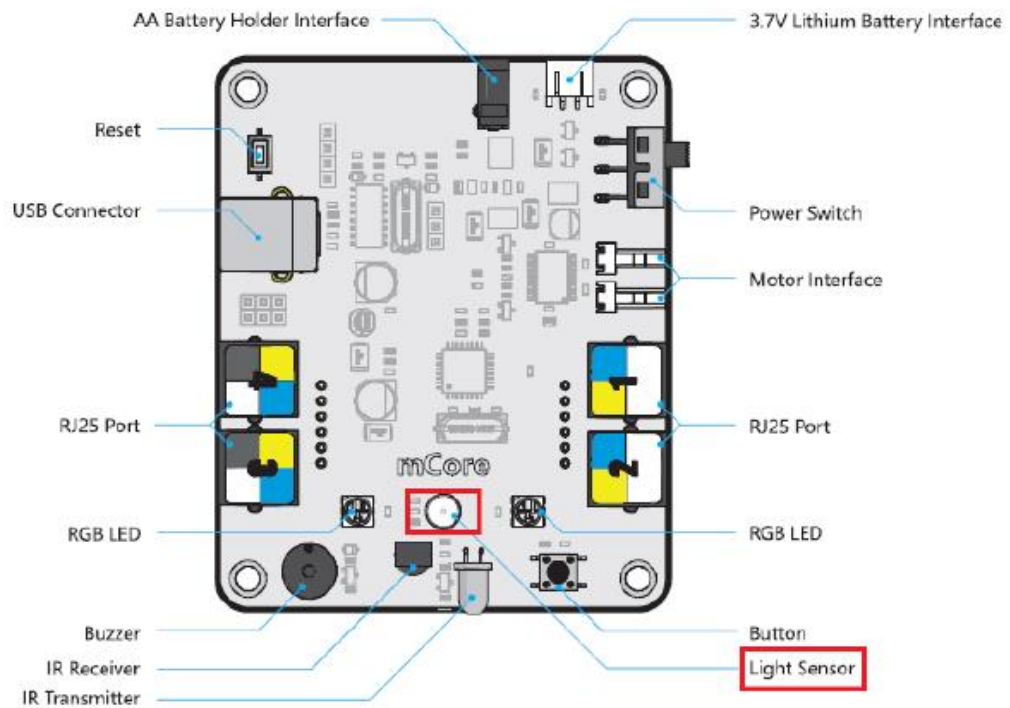
```

when mBot(mcore) starts up
  forever
    if when on-board button pressed ? then
      set a to light sensor on-board light intensity
      LED panel port4 shows text a
  
```

Descrição do código:

Este exemplo explica como capturar valores dos sensores e exibi-los em tempo real. O mesmo método pode ser utilizado para recolher dados de outros sensores.

	<p>Use o bloco "quando o botão integrado PRESSIONAR" para iniciar o programa.</p>
	<p>Use a para armazenar o valor do sensor de luz em tempo real, depois com a matriz LED ligada na porta4 mostramos o valor armazenado na variável</p>
	<p>Este bloco permite usar o bloco para o botão integrado funcionar e passar os dados da leitura de intensidade de luz para a variável.</p>
	<p>Este bloco permite que os valores do sensor de luz sejam exibidos continuamente.</p>



Referências:

<http://www.mblock.cc/example/value-of-the-light-sensor/>



Licença: Creative Commons
 Atribuição não comercial 4.0 Internacional
 (CC BY-NC 4.0)

João Freitas
joapaulofreitas@edu.ulisboa.pt



Projeto:

Comunicação por infravermelhos.

Objetivo:

Quando pressionarmos o botão mBot1 / mCore1 da placa, mBot1 / mCore1 irá enviar sinais de comunicação e de som, e mBot2 / mCore2 receberá o sinal, emitindo sons e luzes.

Código:

Programa 1 - Envio Carregar para mBot1 / mCore1	Programa 2 - Recebendo Carregar para mBot2 / mCore2

Descrição do código:

Este exemplo é implementado usando o mCore recetor de IR e o emissor de IR na placa Arduino.

Prepare dois mBot /mCore, e faça upload do Programa 1 para a mBot1 / mCore1 e do Programa 2 para mBot2 / mCore2.

Programa 1 - O envio de uma mensagem: Basta pressionar o botão da placa mBot1 / mCore1 para enviar uma mensagem, que é enviada pelo emissor infravermelho a bordo. O conteúdo da mensagem pode ser personalizado. Neste exemplo, a mensagem está definida como "A".

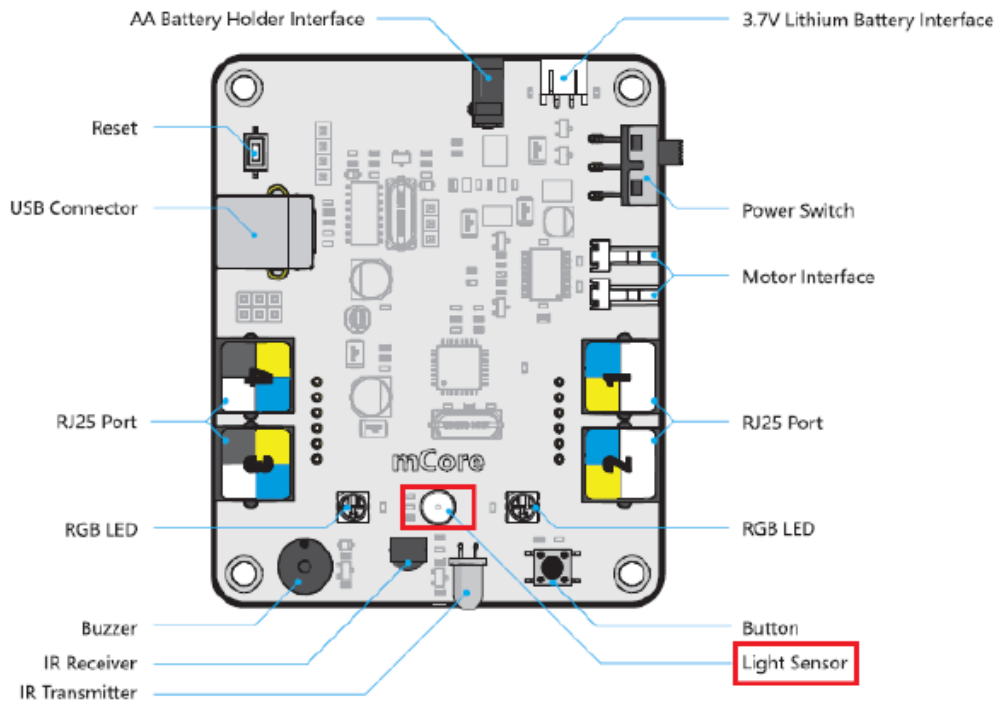
Programa 2 - Receber mensagens: Quando a mensagem "A" é recebida, a placa de controle mBot2 / mCore2 irá acender a luz vermelha e emitir um som (imitando o efeito "hit").

Para implementar o Programa 2, precisamos habilitar os blocos de comunicação.

Usa o bloco  . Selecciona

Desafio:

Tarefa 1	Quando mCore1 sente uma luz forte, ele irá enviar outra mensagem. Quando mCore2 recebe a mensagem, irá mostrar "Luz intensa" com efeitos de som e luz.
----------	--



Referências:

<http://www.mblock.cc/example/infrared-communication/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joaopaulofreitas@edu.ulisboa.pt



Projeto:

Controlar o robô via teclado do computador (o robô deve estar em modo *LIVE* e ligado com cabo USB).

Objetivo:

No teclado do computador, pressiona a tecla ↑ e mBot avança; pressiona a barra de espaços e o mBot acende o LED verde; pressione a tecla número 1, e o mBot toca o alarme.

Código:

```
when up arrow key pressed
  move forward at power 500 %
  wait 1 seconds
  stop moving

when space key pressed
  turn on all light with color red 0 green 60 blue 0
  wait 1 seconds
  turn on all light with color red 0 green 0 blue 0

when 1 key pressed
  play note C4 for 0.25 beats
```

Descrição do código:

	<p>Para usar um teclado para controlar os efeitos de movimento, som e luz de mBot, é necessário usar o bloco da área de "Eventos". Podemos mudar o "espaço" para as setas, barra de espaço, teclas de letras, teclas numéricas e outras teclas do teclado. Neste exemplo, a tecla ↑ é usado para controlar o movimento de mBot. Quando você pressiona a tecla ↑, mBot avança.</p>
--	---



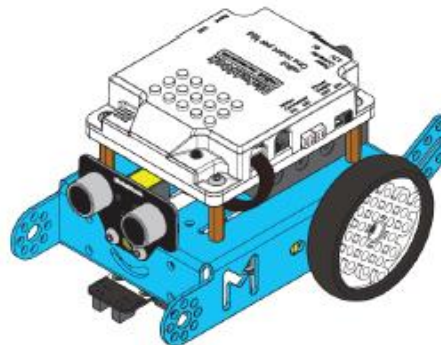
	<p>A potência de rotação em movimento para a frente é de 1 a 100%. Um número positivo representa o movimento para a frente. Quanto maior o número, mais rápido ele se move.</p>
--	---

E se a potência dos motores para avançar é definida com um valor maior que 100%?

Se a velocidade é definida para além de 100, ele será automaticamente definido como 100, que é a velocidade máxima.

Desafio:

Tarefa 1	Reescreva o programa. Pressione somente um botão para movimentar o mBot, fazer luz e som ao mesmo tempo.
----------	--



Referências:

<http://www.mblock.cc/example/infrared-communication/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joaopaulofreitas@edu.ulisboa.pt



Projeto:

Movimentar o mBot em forma de M.

Objetivo:

Pressione o botão da placa do robô e o mBot move-se em forma de M.

Código:

```
when mBot(mcore) starts up
  wait until when on-board button pressed ?
  move forward at power 50 %
  wait 1 seconds
  turn right at power 50 %
  wait 0.5 seconds
  move forward at power 50 %
  wait 1 seconds
  turn left at power 50 %
  wait 0.5 seconds
  move forward at power 50 %
  wait 1 seconds
  turn right at power 50 %
  wait 0.5 seconds
  move forward at power 50 %
  wait 1 seconds
  stop moving
```

Descrição do código:

	Para evitar que o código seja executado logo após a inicialização do mBot. Quando o botão é pressionado, o código será acionado.
--	--



Movimentar o mBot numa pista em forma de M

Dividir a faixa de mBot em 4 segmentos:

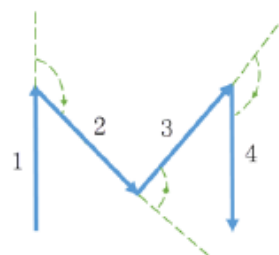
Segmento 1: Mover linear.

Segmento 2: Virar à direita e mover em linha reta.

Segmento 3: Virar à esquerda e movimento em linha reta.

Segmento 4: Virar à direita e mover em linha reta.

Parar o mBot.



Ao alterar a velocidade de do mBot no bloco avançar, virando à esquerda, virando à direita e ajustando a espera, definimos ângulos. Geralmente precisamos testar muitas vezes para obter um bom ângulo. Devemos também considerar as condições do chão e a potência do motor (volume de eletricidade).

Como obter um valor adequado rapidamente

Escolhendo o "Segmento 2: Virar à direita e mover em linha reta", por exemplo, devemos colocar o mBot sobre papel branco e definir um valor apropriado, como seja, virar à direita por 1 segundo a uma potência de 50%. Desenhe no papel branco a direção do mBot. Se o ângulo for muito grande, podemos reduzir o tempo de viragem; se for muito pequeno, podemos aumentar o tempo de viragem. Repita estas etapas até obter um valor adequado.

E se a velocidade de rotação para avançar é definida com um valor maior que 255?

Se a potência é definida para além de 100%, ele será automaticamente definido como 100%, que é a potência máxima.

Desafio:

Tarefa 1	Deixar o mBot movimentar-se noutras faixas como seja em forma de Z.
Tarefa 2	Adicionar luz durante a movimentação em frente do mBot, e buzinar durante a fase em que vira.

Referências:

<http://www.mblock.cc/example/move-along-the-m-shaped-track/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)
João Freitas
joaopaulofreitas@edu.ulisboa.pt



Projeto:

Movimentar o mBot em forma de S.

Objetivo:

Pressione o botão da placa do robô e o mBot move-se em forma de S.

Código:

```

define: Turn left while moving forward
  move forward at power 50 %
  left wheel turns at power 25 %, right wheel at power 50 %

define: Turn right while moving forward
  move forward at power 50 %
  left wheel turns at power 50 %, right wheel at power 25 %

define: Stop
  LED: all shows color green for 2 secs
  stop moving

when mbot(micari) starts up
  wait until when on-board button pressed
  Turn left while moving forward
  wait 2 seconds
  Turn right while moving forward
  wait 2 seconds
  Stop
    
```

Descrição do código:

Neste exemplo, a código esta definido para permitir que mBot se movimente em forma de S.

Chamando os blocos, o programa executa todas as ações definidas dentro deles. Os blocos podem ser reutilizados em várias partes do nosso programa. Desta forma, o programa fica mais simples e mais fácil de ler.

Ações	Comportamento
<pre> move forward at power 50 % left wheel turns at power 25 %, right wheel at power 50 % </pre>	Virar à esquerda enquanto se move em frente
<pre> left wheel turns at power 25 %, right wheel at power 50 % </pre>	Virar à esquerda apenas usando a roda esquerda



	<p>Virar à esquerda apenas usando a roda direita</p>
	<p>Virar à esquerda. O efeito é o mesmo que o Bloco</p>

Desafio:

Tarefa 1	Vamos movimentar o mBot num círculo.
Tarefa 2	Modificar o valor da velocidade de motores esquerdo e direito do mBot (aumentando a diferença entre os dois valores) e verificar o estado de movimento do mBot.

Referências:

<http://www.mblock.cc/example/move-along-the-s-shaped-track/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joopaulofreitas@edu.ulisboa.pt



Projeto:

Simular uma ambulância.

Objetivo:

Pressione o botão da placa do mBot, para ativar a luz de advertência e o som, enquanto se move para a frente, de modo a simular uma ambulância.

Código:

```
when mBot(mcore) starts up
wait until when on-board button pressed ?
move forward at power 30 %
repeat 5
  turn on all light with color red 0 green 0 blue 0
  turn on left light with color red 0 green 0 blue 60
  play sound at frequency of 950 Hz for 0.5 secs
  turn on all light with color red 0 green 0 blue 0
  turn on right light with color red 0 green 0 blue 60
  play sound at frequency of 700 Hz for 0.5 secs
stop moving
```

Desafio:

Tarefa 1	Decore o mBot, por forma a torná-lo parecido com uma ambulância.
----------	--

Referências:

<http://www.mblock.cc/example/simulate-an-ambulance/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joapaulofreitas@edu.ulisboa.pt



Projeto:

Valor do sensor de ultrassom.

Objetivo:

Exibir na matriz LED o valor do sensor de ultrassom.

Código:

```
when mBot(mcore) starts up
  forever
    if when on-board button pressed ? then
      set a to round ultrasonic sensor port3 distance(cm)
      LED panel port4 shows text a
```

Sensor de Ultrassom:

O sensor de ultrassom corresponde aos olhos do mBot, permitindo saber a distância de um objeto que esteja na sua frente. Os objetos podem ser detetados entre um intervalo que vai de um centímetro a 400 centímetros.



Referências:

<http://www.mblock.cc/example/value-of-the-ultrasonic-sensor/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joaopaulofreitas@edu.ulisboa.pt



Projeto:

Automaticamente parar na frente de uma barreira.

Objetivo:

mBot move-se numa certa velocidade e deve parar automaticamente antes de bater num obstáculo.

Código:

```
when mBot(mcore) starts up
forever
  move forward at power 30 %
  wait until ultrasonic sensor port3 distance(cm) < 10
  stop moving
```



Descrição do código:

mBot move-se numa velocidade definida de 150 e pára quando o sensor de ultrassom deteta um objeto a menos de 10 centímetros de distância.

Desafio:

Tarefa 1	Modifica o código para deixar o mBot seguir em frente quando o obstáculo é removido.
Tarefa 2	Se alterarmos o bloco  para o bloco  podemos alcançar o mesmo efeito deste exemplo?

Referências:

<http://www.mblock.cc/example/stop-automatically-in-front-of-a-barrier/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joaopaulofreitas@edu.uisboa.pt



Projeto:

Evitar obstáculos

Objetivo:

O mBot começa a movimentar-se a uma certa velocidade quando o botão é pressionado. Quando há uma barreira na frente, o mBot vai virar à direita; quando não há, o mBot continua movendo-se em frente.

Código:

```

when mBot(mcore) starts up
wait until when on-board button pressed ?
forever
if ultrasonic sensor port3 distance(cm) > 10 then
move forward at power 50 %
else
turn right at power 50 %
wait 1 seconds
    
```




Descrição do código:

Use o botão da placa para movimentar o mBot em frente. O valor do sensor de ultrassom indica a distância entre o mBot e qualquer obstáculo que esteja à frente dele. Dado o limite de 10cm, que indica que não há nenhuma barreira na frente do mBot a menos de 10 cm, sendo que nestas condições o seu movimento segue em frente; quando existe um obstáculo a menos de 10cm, o mBot precisa de mudar de direção para evitar o objeto.



Neste exemplo indica que, se é verdade, o primeiro bloco de instruções é executado, caso contrário, se é false, será o segundo bloco de instruções a ser executado.

Desafio:

Tarefa 1	Modificar o código e usar o bloco  para conseguir evitar o obstáculo.
Tarefa 2	Alterar a ação do mBot para que quando encontre um obstáculo mova para trás e contorne o obstáculo.



Referências:

<http://www.mblock.cc/example/avoid-barriers/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joopaulofreitas@edu.ulisboa.pt



Projeto:

Sensor de linha.

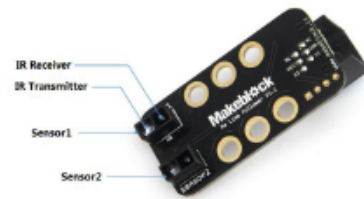
Objetivo:

A matriz LED indica o valor do sensor de linha em tempo real.

Código:

```

when mBot(mcore) starts up
  forever
    if when on-board button released ? then
      set a to line follower sensor port2 value
      LED panel port4 shows text a
  
```



Sensor de linha:

O sensor de linha situa-se na parte de baixo do robô (ver diagrama anexo), que é constituído por dois sensores, o sensor 1 e 2, cada um composta por um emissor e um recetor de infravermelhos (ver o diagrama anexo). Como muitas vezes é usado para manter o robô se movendo em linha reta, ele é chamado de sensor de linha. A sua gama de deteção é de 1 a 2 cm.

O emissor infravermelho emite continuamente luz infravermelha durante o movimento do mBot:

Se a luz infravermelha é refletida (encontrando superfícies de cores de luz branca ou outros), o recetor recebe o sinal infravermelho, sendo o valor de saída de 1 (podemos ver que o LED azul na parte de trás do sensor de linha é iluminado); Se a luz infravermelha é absorvida ou não pode ser refletida, o recetor não recebe o sinal de infravermelhos sendo o valor de saída de 0.

De acordo com a tabela seguinte, o valor do sensor de linha pode assumir apenas quatro valores: 0, 1, 2, e 3.

	Se o recetor recebe o sinal de infravermelhos		Valor retornado pelo Sensor 1	Valor retornado pelo Sensor 2	Representado em binário	Binário para decimal (o valor mostrado na fase pelo sensor)
	Sensor1	sensor2				
situação 1	Não recebido	Não recebido	0	0	00	0
situação 2	Não recebido	Recebido	0	1	01	1



situação 3	Recebido	Não recebido	1	0	10	2
situação 4	Recebido	Recebido	1	1	11	3

Desafio:

Tarefa 1	Coloque um papel preto e um papel branco em frente do sensor de linha, respetivamente, e observe, no palco do mBlock, o valor mostrado.
----------	---



Referências:

<http://www.mblock.cc/example/avoid-barriers/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joaopaulofreitas@edu.uisboa.pt



Projeto:

Coloca-me em cima de uma superfície clara.

Objetivo:

Coloque mBot numa mesa de cor clara. Uma vez mBot é levantado, soa um alarme continuamente e o valor do sensor de linha é mostrado na matriz LED em tempo real.

Código:

```
when mBot(mcore) starts up
  forever
    if line follower sensor port2 value = 0 then
      play note A4 for 0.25 beats
    set a to line follower sensor port2 value
    LED panel port4 shows text a
```

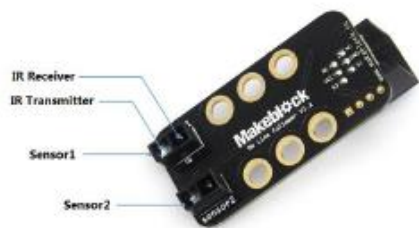
Descrição:

Quando mBot é colocado sobre uma mesa branca, a luz infravermelha pode ser refletida, e o valor é de 3 (tanto Sensor1 e Sensor2 saída 1); quando mBot é levantado em cerca de 1-2 cm a partir da superfície da mesa, excede o alcance de deteção, a luz infravermelha não pode ser refletida e o valor é 0 (ambos Sensor1 e Sensor2 ficam com saída de 0).

Quando o valor do sensor de linha-patrolha é 0, pode ser considerado que mBot foi levantado e soa o alarme. Através do Módulo LED Matrix é possível de ver o valor de saída do sensor

Desafio:

Tarefa 1	Certifique que o mBot soa um alarme quando é levantado e mostre no palco o valor do sensor de linha.
----------	--



Referências:

<http://www.mblock.cc/example/put-me-down/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)

João Freitas
joaopaulofreitas@edu.uisboa.pt



Projeto:

Detetar um penhasco

Objetivo:

Coloque mBot numa mesa de cor clara. Uma vez mBot é levantado, soa um alarme continuamente.

Código:


```
when mBot(mcore) starts up
  forever
    if line follower sensor port2 value = 0 then
      move forward at power 0 %
    else
      move forward at power 30 %
```

Descrição:

Quando mBot viaja no interior da superfície branca da mesa (ou outra tabela de cores de luz), o sensor de linha pode receber o sinal de infravermelhos, sendo o seu valor de 3 (tanto o Sensor1, como o Sensor2 retornam 1); quando mBot viaja para a borda da superfície da mesa, o sensor de linha não pode receber a luz infravermelha refletida e o valor é 0 (ambos Sensor1 e Sensor 2 retornam 2). Quando o valor do sensor de linha é 0, é considerado como tendo atingido a borda da superfície da mesa e irá deixar de se movimentar.

Além disso, se usarmos este programa, o mBot vai parar de se mover para a frente quando deteta a cor preta.

Desafio:

Tarefa 1	Modifica o código para deixar o mBot voltar para trás quando deteta a borda da superfície da mesa.
Tarefa 2	Modifica o código, por forma a usar bloco  .

Referências:

<http://www.mblock.cc/example/cliff-detection/>



Licença: Creative Commons
Atribuição não comercial 4.0 Internacional
(CC BY-NC 4.0)
João Freitas
joaopaulofreitas@edu.ulisboa.pt



Projeto:

Seguidor de linha.

Objetivo:

Movimentar o mBot ao longo de uma faixa preta numa superfície de cor clara.

Código:

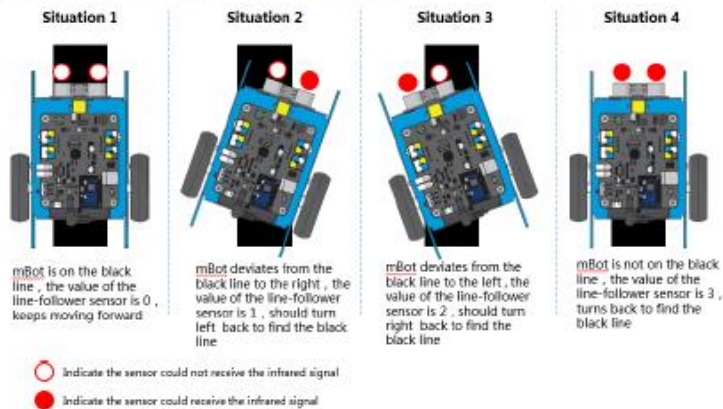
```

when mBot(more) starts up
wait until [when on-board button pressed]
forever
if [line follower sensor port2 value = 0] then
  move forward at power 30 %
if [line follower sensor port2 value = 1] then
  left wheel turns at power 0 %, right wheel at power 30 %
if [line follower sensor port2 value = 2] then
  left wheel turns at power 30 %, right wheel at power 0 %
if [line follower sensor port2 value = 3] then
  move backward at power 30 %
    
```

Descrição:

O mBot irá movimentar-se numa faixa de cor preta, desenhada numa superfície de cor clara.

Neste exemplo temos quatro situações em que mBot deve tomar decisões:





```

when mBot(mcore) starts up
wait until [when on board button pressed]
forever
  if [line follower sensor port2 value = 0] then
    move forward at power 30 %
  if [line follower sensor port2 value = 1] then
    left wheel turns at power 0 %, right wheel at power 30 %
  if [line follower sensor port2 value = 2] then
    left wheel turns at power 30 %, right wheel at power 0 %
  if [line follower sensor port2 value = 3] then
    move backward at power 30 %
  
```

Situação 1

Situação 2

Situação 3

Situação 4

Desafio:

Tarefa 1	Alterar a velocidade do mBot quando ele se está a mover para a frente ou a virar. Observar os efeitos do sensor de linha do mBot.
Tarefa 2	Vamos pensar: na situação 4, que estratégia pode o mBot usar para encontrar a linha preta.

Referências:

<http://www.mblock.cc/example/primary-line-patroling-program/>



Licença: Creative Commons
 Atribuição não comercial 4.0 Internacional
 (CC BY-NC 4.0)

João Freitas
joaopaulofreitas@edu.ulisboa.pt